Siril

Release 1.4.4

Free-Astro Team

04.10.2026

Allgemeines

1 Installation **3**
1.1 Mindest-Systemanforderungen . . . 4
1.2 OS spezifische Installation . . . 4

2 Grafische Benutzeroberfläche **15**
2.1 Haupt-Benutzerinterface . . . 15
2.2 Main Toolbar . . . 25
2.3 Hamburger-Menü . . . 27
2.4 Werkzeugmenü . . . 27
2.5 Bildbearbeitungsmenü . . . 28
2.6 Verarbeitung von Teilbereichen . . . 28
2.7 Fenster mit Bildinformationen . . . 31
2.8 Tastenkürzel . . . 32

3 Arbeitsverzeichnis **35**

4 Voreinstellungen **37**
4.1 Einstellungen (GUI) . . . 37
4.2 Einstellungen (Befehle) . . . 64

5 Dateiformate **75**
5.1 Bittiefe . . . 75
5.2 Gebräuchliche Dateiformate . . . 75
5.3 FITS Format . . . 80
5.4 Astro-TIFF . . . 86
5.5 SER Format . . . 90
5.6 IRIS PIC . . . 91
5.7 PixInsight XISF . . . 91

6 Farbmanagement **93**
6.1 Erste Schritte . . . 93
6.2 Status des Farbmanagements . . . 96
6.3 Farbverwaltungsdialog . . . 97
6.4 ISO 12646 Bildbewertungsmodus . . . 99
6.5 Befehle . . . 99
6.6 Farbkonvertierungsmatrix . . . 100
6.7 Theorie der Farbverwaltung . . . 103

6.8 Arbeitsablauf . . . 108
6.9 Farbraumprofile . . . 110
6.10 Soft-Proofing . . . 112
6.11 Änderungen durch die Einführung von Farbmanagement . . . 114

7 Sequenzen **117**
7.1 Ein Satz aus zwei oder mehr FITS-Dateien . . . 117
7.2 Eine einzelne SER-Datei . . . 118
7.3 Eine einzelne FITS-Datei . . . 118
7.4 Eine Sequenz laden . . . 118
7.5 Bilderliste . . . 119
7.6 Sequenzexport . . . 121
7.7 Sequenzinformationen . . . 122

8 Definitionen und Arbeitsablauf **123**

9 Vorberechnung **125**
9.1 Umrechnung . . . 125
9.2 Kalibrierung . . . 131
9.3 Registrierung . . . 157
9.4 Drizzle . . . 178
9.5 Stacken . . . 189

10 In Bearbeitung **197**
10.1 Stretching der Bilder . . . 197
10.2 Farbkalibrierung . . . 217
10.3 Andere Farbwerkzeuge . . . 238
10.4 Filter . . . 244
10.5 Sternbearbeitung . . . 291
10.6 Geometrie . . . 301
10.7 Hintergrund-Extraktion . . . 306
10.8 GraXpert Schnittstelle . . . 311
10.9 Kanalextraktion . . . 317
10.10 Lineare Übereinstimmung . . . 321
10.11 RGB-Zusammensetzung . . . 321
10.12 CFA Kanäle zusammenführen . . . 326
10.13 Pixel Math . . . 329

11 Astrometrie **337**
11.1 Platesolving/astrometrische Lösung . . . 337
11.2 Beschriftungen/Objektnamen . . . 354

12 Dynamische Point Spread Funktion **369**
12.1 Erste Sternkandidaten . . . 370
12.2 Modelle . . . 370
12.3 Minimierung . . . 374
12.4 Nutzung . . . 375
12.5 Konfiguration . . . 377
12.6 Quellenverzeichnis . . . 379

13 FITS Header **381**

14 Intensitätsprofilierung **387**
14.1 Basis-Intensitätsprofil . . . 387
14.2 Profiltypen . . . 388

14.3 Präzise Koordinateneingabe . . . 391
14.4 Messungen . . . 391
14.5 Siril Grafikwerkzeug . . . 391
14.6 Befehle . . . 392

15 Bildinspektoren **395**
15.1 Verkippung . . . 395
15.2 Inspektor für die Bildfeldwölbung (Abberration Inspektor) . . . 397

16 Photometrie **399**
16.1 Grundlagen . . . 400
16.2 Schnelle Photometrie . . . 403
16.3 Lichtkurven . . . 411

17 Plot-Funktion **429**
17.1 Plot/Grafische Darstellung der Registrierungsdaten . . . 429
17.2 Photometrische Grafik . . . 433
17.3 Gemeinsame Optionen . . . 435
17.4 Grafik-Interaktionen . . . 435

18 Statistik **437**
18.1 Abschätzungen . . . 439

19 Siril für Wissenschaftler **441**
19.1 How can Siril help me with my research? . . . 441
19.2 Which version of Siril should I install? . . . 441
19.3 How can I get in touch with the developers? . . . 441
19.4 How accurate are Siril results? . . . 442
19.5 How can I cite Siril? . . . 442
19.6 How can I open FITS cubes with Siril? . . . 442
19.7 How do I measure FWHM in an image? . . . 442
19.8 How do I use star detection (source extraction)? . . . 443
19.9 Welche Astrometrie-Tools stehen zur Verfügung? . . . 443
19.10 Kann es Objekte des Sonnensystems verarbeiten? . . . 443
19.11 Kann Siril mit Spektren arbeiten? . . . 443
19.12 Can Siril replace IRAF? . . . 443

20 Skripting **445**
20.1 SIRIL Skriptdateien . . . 445
20.2 Pythonskripte . . . 453
20.3 Python Script Key Information for Users . . . 459
20.4 Python Script Key Information for Authors . . . 460
20.5 Skripteditor . . . 464
20.6 Python Skriptbeispiele . . . 466
20.7 Current list of Python scripts for Siril 1.4 . . . 484
20.8 Skripte laden . . . 487

21 Headless Modus **489**
21.1 Befehlsstrom (Pipe) . . . 490

22 Pfadnamen Parsing **491**
22.1 Beispiel für die Syntax . . . 491
22.2 Auffinden einer Datei mit Pfadanalyse . . . 494
22.3 Schreiben einer Datei mit Dateinamen-Parsing . . . 494
22.4 Verwendung von Platzhaltern/Wildcards . . . 495

22.5 Umgang mit Duplikaten . . . 496

23 Live-Stacking **497**
23.1 Live-Stacking (GUI) . . . 497
23.2 Live-Stacking (ohne GUI/Headless) . . . 499

24 Befehle **501**

25 Sirilpy Python Module API 1.0.25 Reference **575**
25.1 Abhängigkeiten . . . 576
25.2 Siril Script Coding Guidance . . . 576
25.3 Sirilpy Connection . . . 577
25.4 Sirilpy Datenmodelle . . . 602
25.5 Sirilpy Enums . . . 619
25.6 Sirilpy Plotting . . . 625
25.7 Sirilpy Utility Methoden . . . 626
25.8 Sirilpy GPU Helfer . . . 630
25.9 Sirilpy TK GUI . . . 635
25.10 Sirilpy Tkfilebrowser . . . 637
25.11 Sirilpy Exceptions . . . 638

26 Wie man Probleme meldet **641**
26.1 Änderungsprotokolle und Bug-Tracker prüfen . . . 641
26.2 Senden Sie uns nützliche Informationen . . . 641
26.3 Wie können Sie uns kontaktieren? . . . 642

Literaturverzeichnis **645**

Python-Modulindex **647**

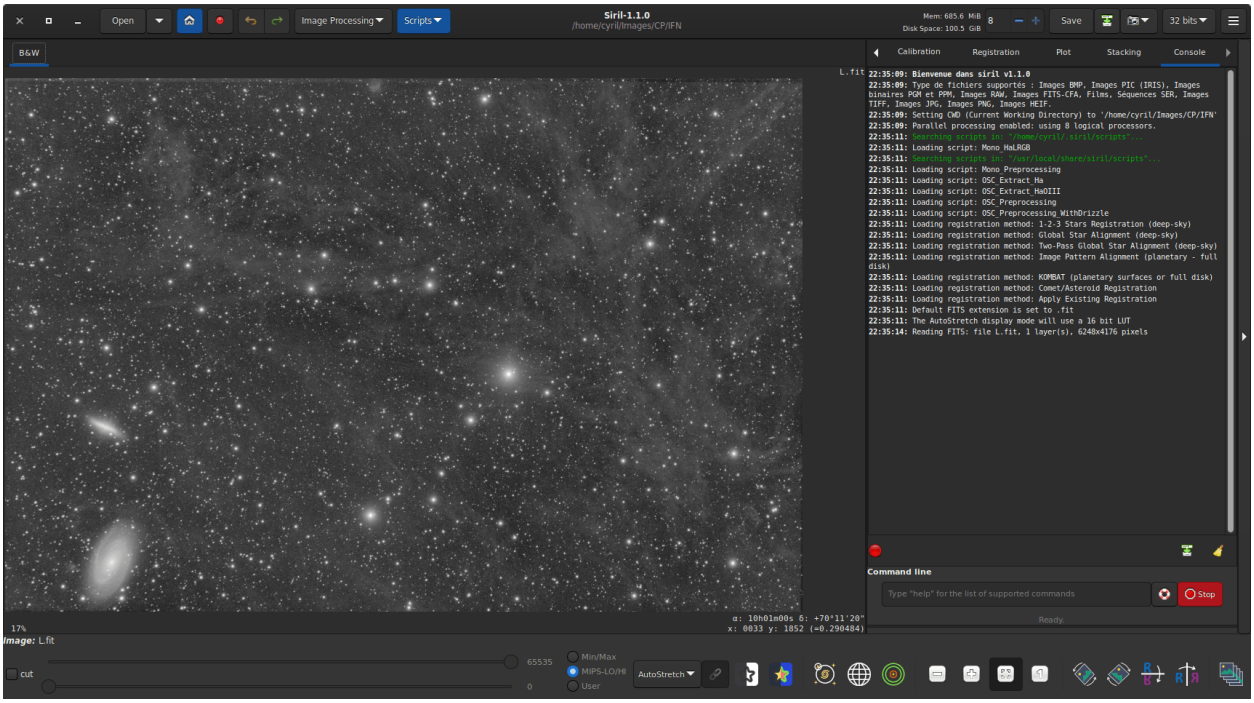
Commands Index **649**

Stichwortverzeichnis **653**

JOSS 10.21105/joss.07242

Dies ist die Dokumentation der Version 1.4.4.

Siril is a deep sky astronomical image processing tool.



Siril kann Bilder aus verschiedenen Dateiformaten kalibrieren, ausrichten, stapeln und verbessern, sogar Bildsequenzdateien (FITS-Cube, Filme und SER-Dateien). Siril kann Bilder auch mit Astrometrie- und Photometrie-Tools analysieren.

Siril ist weitgehend in der Programmiersprache C geschrieben, Teile in C++. Die Hauptentwicklung wird mit den aktuellsten Versionen von Funktionsbibliotheken auf GUN/Linux durchgeführt. Mitwirkende sind willkommen.

Dies ist die Dokumentation, sie versucht, alle Siril-Funktionen zu beschreiben. Wenn es eine Entsprechung zu einer GUI-Funktion für die Kommandozeile gibt, dann wird sie in einem Einschub angegeben. Weitere nützliche Ressourcen finden Sie auf unserer Haupt-Website siril.org

Tipp

Diese Dokumentation enthält auch verschiedene Kästen mit **Tipps**- und **Warnungen**, die wichtige Informationen hervorheben. Darüber hinaus gibt es **Theorie**-Kästen, die für das Verständnis des Inhalts nicht unbedingt erforderlich sind, aber für diejenigen mit mathematischem Hintergrund verfügbar sind, die neugierig sind und tiefer in bestimmte Konzepte eintauchen möchten.

Hier finden einen Index der Siril Befehle.

Um Probleme in der Dokumentation zu melden, öffnen Sie bitte ein Ticket unter der folgenden Adresse: https://gitlab.com/free-astro/siril-doc.

Ein Problem in der Übersetzung der Dokumentation sollte hier gemeldet werden: https://gitlab.com/free-astro/siril-localized-doc.

KAPITEL 1

Installation

Jede Version von Siril wird für die drei meistgenutzten Plattformen (Windows, MacOS, GNU/Linux) bereitgestellt und kann auf der Siril Webseite heruntergeladen werden. Aber natürlich können Sie die Applikation auch aus den Quelltexten erstellen, das Siril eine offene, freie Software ist.

Tipp

Es kann nützlich sein, die Integrität der Binärdatei oder des Pakets zu überprüfen, das Sie gerade heruntergeladen haben. Die Liste der SHA-Prüfsummen ist auf dieser Seite im json-Format verfügbar.

Nach der Installation können Sie den *capabilities*-Befehl nutzen um mehr über Ihre Installation zu erfahren.

Siril Kommandozeile

```
capabilities
```

Auflistung der Siril-Fähigkeiten, basierend auf den Kompilierungsoptionen und Laufzeit

Die Versionsnummer von Siril verstehen

Beginnend mit der Version 1.0 sind stabile Versionen von Siril (wie 1.0, 1.2 usw.) durch gerade Zahlen gekennzeichnet und für den täglichen Gebrauch bestimmt. Entwicklungsversionen, die durch ungerade Zahlen gekennzeichnet sind (z. B. 0.99.0, 1.1.0 usw.), sind normalerweise nicht als Pakete oder ausführbare Binärdateien verfügbar und müssen vom Benutzer kompiliert werden. Die dritte und letzte Zahl, die so genannte Mikronummerierung, entspricht der Anzahl der Veröffentlichungen, die Fehlerkorrekturen und andere kleine Beiträge enthalten (z. B. 1.0.1, 1.0.2, 1.0.3, usw.).

1.1 Mindest-Systemanforderungen

Um eine optimale Leistung zu gewährleisten, empfiehlt es sich, dass Ihr System die folgenden Mindestanforderungen erfüllt:

- **RAM**: Mindestens 8 GB RAM werden für alle Systeme empfohlen, um einen reibungslosen Betrieb zu gewährleisten. Eine höhere RAM-Kapazität ist von Vorteil, da Siril den verfügbaren Arbeitsspeicher umfassend für die Verarbeitung von Aufgaben nutzt.
- **Speicherung**: Ein SSD (Solid State Drive) wird für schnellere Lese-/Schreibgeschwindigkeiten dringend empfohlen, was im Vergleich zu herkömmlichen Festplatten zu einer **um eine Größenordnung schnelleren Ausführung** führt

Die spezifischen Anforderungen an die Plattform sind wie folgt:

- **Windows**: Windows 8.1 oder neuer.
- **macOS**: macOS 13 Ventura or later.
- **GNU/Linux**: Keine spezifischen Mindestanforderungen angegeben. Normalerweise sollte jede moderne Distribution ausreichen.

1.2 OS spezifische Installation

The following pages explain installation procedures for the different OSes.

1.2.1 Installation auf GNU/Linux

Installation unter Debian

Das Binärpaket ist auf Debian testing und in einer alten Version unter stable verfügbar. Es kann über apt mit Superuser-Rechten installiert werden:

Installation auf Ubuntu oder Linux Mint

Offizielle Repositories

Wie für Debian ist es in den Repositories verfügbar, jedoch kann die Version veraltet sein:

```
sudo apt install siril
```

PPA Repositories

Die neueste Version ist daher in unserem PPA verfügbar, was die bevorzugte Methode zur Installation von Siril unter Ubuntu oder Linux Mint ist:

```
sudo add-apt-repository ppa:lock042/siril
sudo apt-get update
sudo apt-get install siril
```

Installation der Applikations-Binärdatei

Für GNU/Linux-Systeme haben wir uns außerdem entschieden, gebündelte Binärdateien mit AppImage (x86_64) und Flatpak bereitzustellen, die auf GNU/Linux-ähnlichen Systemen funktionieren. Um die AppImage-Binärdatei auszuführen, müssen Sie sie nur herunterladen und mit dem folgenden Befehl ausführen:

```
chmod +x Path/To/Application/Siril-x.y.z-x86_64.AppImage
```

Durch Ersetzen mit dem richtigen Pfad und x, y und z mit den Versionsnummer. Dann startet ein einfacher Doppelklick auf das AppImage Siril.

Installation von flatpak

Eine andere Möglichkeit, eine stabile Version von Siril zu installieren, ist die Verwendung von flatpak, dem Dienstprogramm für die Softwareverteilung und Paketverwaltung für Linux. Um Flatpak zu installieren, geben Sie den folgenden Befehl ein:

```
flatpak install flathub org.siril.Siril
```

Dann führen Sie die Anwendung aus:

```
flatpak run org.siril.Siril
```

1.2.2 Installation unter Microsoft Windows

Installation mit dem Installationsprogramm

Es wird empfohlen, Siril mit dem mitgelieferten Installationsprogramm zu installieren, das Sie Schritt für Schritt anleitet.

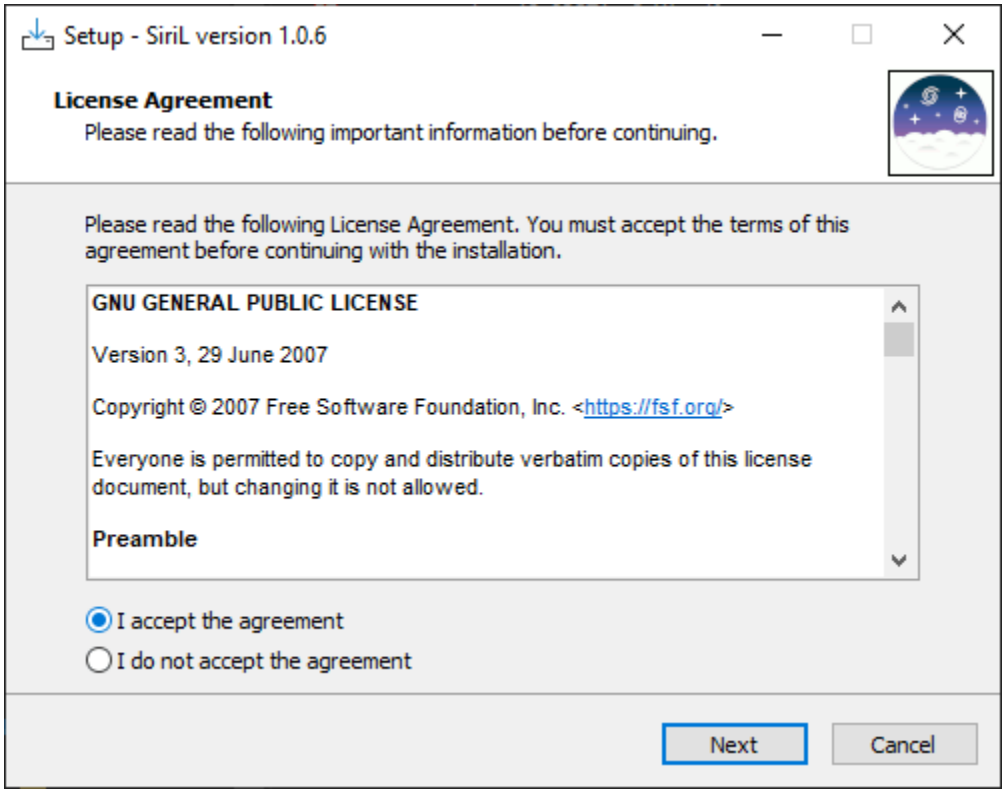


Abb. 1: Auf dem ersten Bildschirm des Installationsprogramms müssen Sie die Vereinbarung akzeptieren, um fortzufahren.

Der Siril-Setup-Assistent installiert alle erforderlichen Dateien an der richtigen Stelle und am Ende haben Sie die Wahl, ob Sie eine Verknüpfung auf dem Desktop erstellen möchten oder nicht.

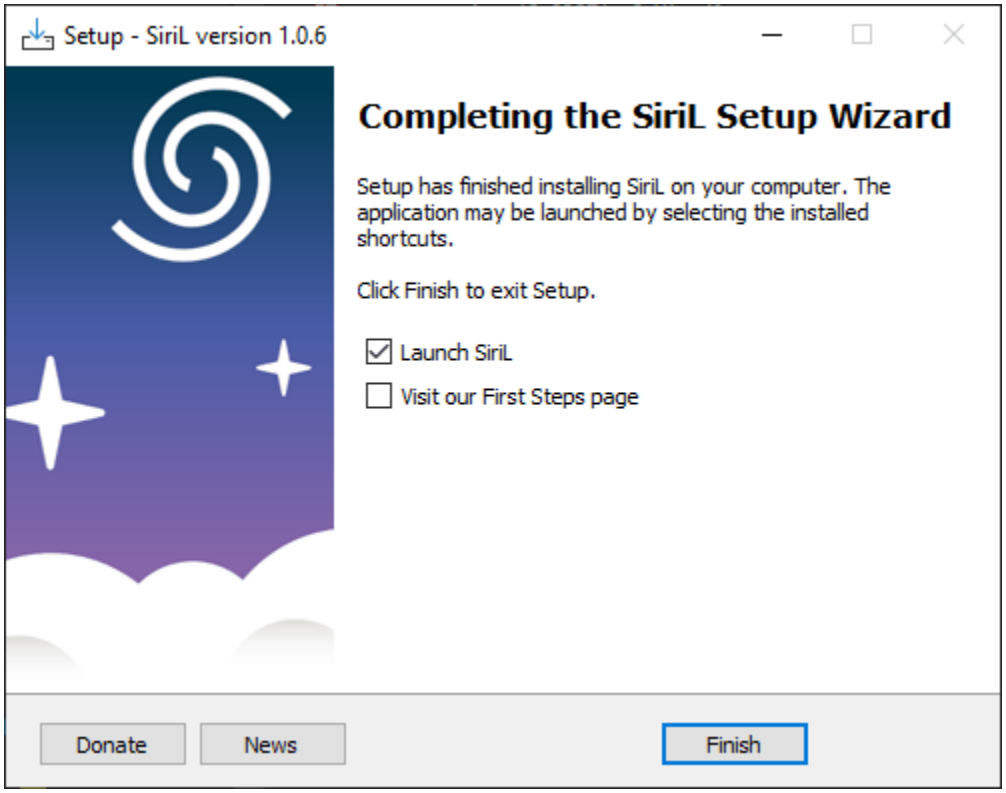


Abb. 2: Letzter Bildschirm des Installationsprogramms. Sie können wählen, ob Sie Siril direkt nach der Installation starten und das Tutorial öffnen möchten, das die ersten Schritte erklärt.

Bemerkung

Siril wird in C:\Programme\Siril installiert. Wenn Sie nicht die Rechte haben, in diesen Ordner zu installieren, verwenden Sie stattdessen eine portable Version (siehe *Installation der portablen Binärdatei*.)

Installation der portablen Binärdatei

Wenn Sie Siril verwenden möchten, ohne alle möglichen Dateien auf Ihrem Computer zu installieren (zum Beispiel, wenn Sie keine Administratorrechte auf dem Rechner haben), dann ist es empfehlenswert, die portable Version zu verwenden. Sie wird in Form einer Zip-Datei geliefert, die Sie einfach an einem Ort Ihrer Wahl entpacken und dann in den Ordner `bin` gehen, um `siril.exe` auszuführen. Sie können auch eine Verknüpfung auf Ihrem Desktop erstellen, um den Start der Anwendung zu erleichtern.

Warnung

Seien Sie vorsichtig, unter keinen Umständen sollten Sie die `exe`-Datei oder irgendeine andere Datei verschieben. Sonst wird Siril nicht laufen.

Kompilieren unter Windows mit Msys2

These instructions are made for compiling on Windows with MSYS2 using the UCRT64 toolchain. MSYS2 requires 64-bit Windows 10 or newer, and does not work with FAT filesystems.

Bemerkung

Starting with Siril 1.4.2, UCRT64 is the only supported build environment on Windows. The MinGW64 environment, previously supported as an alternative, has been deprecated by the MSYS2 project on March 15th, 2025. Several Siril dependencies have already been removed from the MinGW64 package index, and more are expected to follow. All future releases of Siril will be built exclusively with UCRT64.

If you are using the installer or portable binaries provided on the Siril website, this change is transparent to you. If you are building Siril from sources, you must use the UCRT64 environment.

Using UCRT64 also increases the number of files that can be handled in a sequence to 8192.

Download MSYS2 64bit, a software distribution and building platform for Windows, and run the `x86_64` installer for 64-bit. When asked, specify the directory where MSYS2 will be installed.

Run MSYS2 directly from the installer, or later **MSYS2 MinGW UCRT x64** from the Start menu or a shortcut.

Warnung

Make sure to launch the UCRT x64 shell (the icon at the top of the terminal window should be yellow).

First, update the package database and core system packages by typing (for more information about pacman, see this page):

```
pacman -Syu
```

Installieren von Abhängigkeiten

Um die Abhängigkeiten zu installieren, geben Sie den folgenden Befehl ein:

```
curl -s -o siril-deps.sh https://gitlab.com/free-astro/siril/-/raw/master/build/windows/
↪native-gitlab-ci/siril-deps.sh
bash siril-deps.sh
```

Bemerkung

The link above points to the latest version of the dependencies installation script. We have had a tendency in the past not to remove any dependencies (rather to add more), so even if you want to build the stable branch, you should be okay using the latest dependencies script. In case a dependency is missing for a specific version, you can always check the script from the corresponding tag in the GitLab repository.

Erstellen aus den Quelldateien

The source code is hosted on GitLab. Download it with the following command:

```
git clone https://gitlab.com/free-astro/siril.git
cd siril
git submodule update --init
```

Generate the build system and compile the code:

```
meson setup _build --buildtype release
ninja -C _build install
```

To launch your build of Siril, open an MSYS2 UCRT64 shell and type:

```
siril
```

You can also create a shortcut to `siril.exe`, located by default at `C:\msys2\ucrt64\bin\`.

To update your version, open an MSYS2 UCRT64 shell, then:

```
pacman -Syu
cd siril
git pull --recurse-submodules
meson setup _build --reconfigure
ninja -C _build && ninja -C _build install
```

If `git pull` does not show any changes, there is no need to rebuild. Otherwise, the above commands will update your build. Then launch Siril with:

```
siril
```

Warnung

Starting with Siril 1.3.6, you will need to have Python installed on your system if you build from sources. The first time you launch a self-built Siril, you must do so from a native Windows environment, not from the MSYS2 shell, in order for Siril to detect the Windows Python installation rather than the MSYS2 one (a notification will appear

in the console if this step is skipped, and the sirilpy module will not be initialized). This is a one-time requirement. Once the virtual environment has been set up, you can start Siril from the MSYS2 shell as usual.

To launch Siril in a Windows environment, either open a Windows terminal and run `C:\msys2\ucrt64\bin\siril.exe`, or locate this file in Explorer and double-click it.

1.2.3 Installation unter macOS

Installation unserer App

Die macOS-App wird pro Architektur bereitgestellt:

- Intel (macOS 13 Ventura+)
- Apple Silicon (macOS 13 Ventura+)

Wählen Sie den Link, der Ihrer Prozessorarchitektur entspricht, und laden Sie das Disk-Image herunter. Doppelklicken Sie nach dem Herunterladen auf das Image, um es zu öffnen.

Ein neues Fenster wird geöffnet. Ziehen Sie nun das `Siril`-Icon auf das `Applications`-Icon.

Herzlichen Glückwunsch, Siril ist jetzt installiert.

Installation mit Homebrew

Homebrew ist ähnlich wie MacPorts und bietet Pakete (auch Formeln genannt) zur Installation an, entweder durch Kompilieren aus dem Quellcode oder durch Verwendung vorkompilierter Binärdateien (auch Flaschen genannt). Um Homebrew zu installieren, klicken Sie hier. Siril kann installiert werden mit:

```
brew install siril
```

Bemerkung

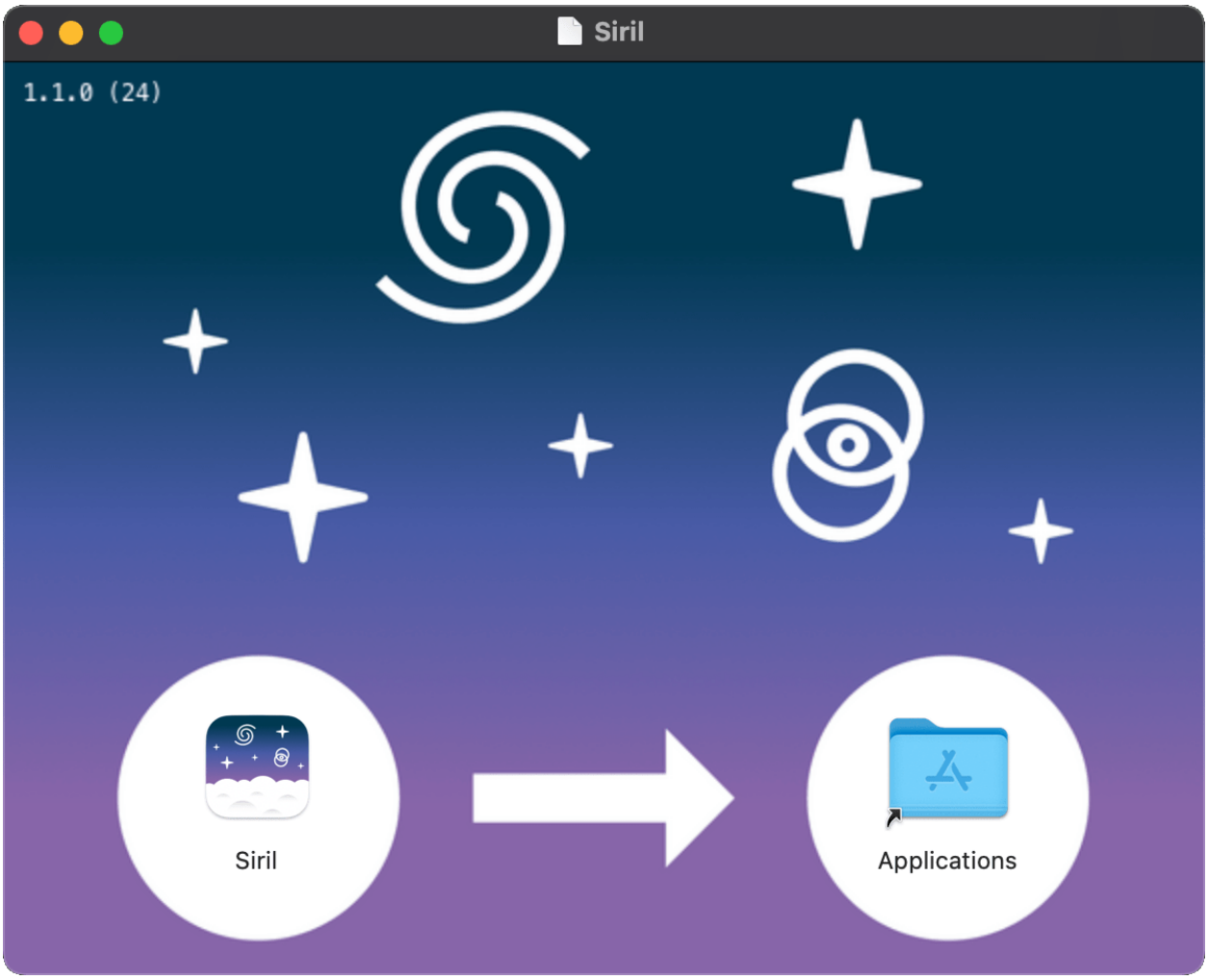
Bitte beachten Sie, dass angekündigt wurde, dass Homebrew Analytics verwendet. Um dies zu deaktivieren, führen Sie aus: `brew analytics off` Sie können mehr darüber auf Brew Analytics lesen.

1.2.4 Installation aus den Quelldateien

Die Installation aus dem Quellcode ist erforderlich, wenn Sie die neuesten Funktionen nutzen möchten, wenn die vorherige Version veraltet ist, wenn Sie an der Verbesserung von Siril teilnehmen möchten oder nicht alle Abhängigkeiten verwenden möchten.

Beschaffung der Quelldateien

Die Quelldateien werden in einem Git-Repository gespeichert, das Sie beim ersten Mal mit diesem Befehl herunterladen können:


Siril
1.1.0 (24)
Siril
Applications

```
git clone --recurse-submodules https://gitlab.com/free-astro/siril.git
```

Aktualisieren Sie es regelmäßig durch Eingabe von `git pull` im Siril-Basisverzeichnis:

```
git pull
git submodule update --recursive
```

Abhängigkeiten

Siril ist auf eine Reihe von Bibliotheken angewiesen, von denen die meisten in der Linux-Distribution oder im Paketmanager Ihrer Wahl verfügbar sein sollten. Die Namen der betriebssystemspezifischen Pakete sind in den einzelnen Abschnitten unten aufgeführt. Obligatorische Abhängigkeiten sind:

- gtk+3 (Bibliothek für die grafische Benutzeroberfläche), mindestens Version 3.20
- adwaita-icon-theme (Icons) um das "look and feel" der Benutzeroberfläche zu unterstützen.
- cfitsio (Unterstützung des FITS-Dateiformats).
- fftw (Bibliothek zur diskreten Fourier Transformation).
- gsl (The GNU Scientific Library), Version 1 oder 2 beginnend mit Siril 0.9.1 or SVN Revision 1040.
- OpenCV und ein C++ Compiler für einige Funktionen der Bildbearbeitung.
- Little CMS ein Open-Source Farbverwaltungssystem
- wcslib für die Verarbeitung des Welt-Koordinatensystems (WCS), Objektnamen und photometrische Farbkalibrierung.
- gtksourceview4 for multiline text editing. GtkSourceView adds support for syntax highlighting, undo/redo, file loading and saving, search and replace, a completion system, printing, displaying line numbers, and other features typical of a source code editor.

Bemerkung

Auch wenn Siril seit Version 0.9.9 in der Konsole laufen kann, ist es immer noch gegen die grafischen Bibliotheken gelinkt, so dass Sie immer noch GTK++ benötigen, um es zu kompilieren und auszuführen.

Optionale Abhängigkeiten sind:

- openmp für Multithreading. Obwohl optional, wird diese Abhängigkeit dringend empfohlen, da die Leistung dadurch wesentlich besser ist. Das Flag dieser Option ist standardmäßig auf true gesetzt. Das heißt, wenn openmp nicht auf Ihrem Computer installiert ist, müssen Sie „-Dopenmp=false“ im Meson-Setup hinzufügen.
- libraw, libtiff, libXISF, libjpeg, libjxl, libpng, libheif for RAW, TIFF, XISF, JPEG, JPEG XL, PNG, HEIF and AVIF images import and export. The libraries are detected at compilation-time.
- FFMS2 für die native Unterstützung von Videodateien als Bildsequenzen. Es ermöglicht auch die Extraktion von Einzelbildern aus vielen Arten von Videodateien, auch für andere Zwecke als Astronomie. Versionen <2.20 haben einen lästigen Fehler. Es wird empfohlen, die neueste Version zu installieren.
- ffmpeg (or libav), providing libavformat, libavutil (>= 55.20), libavcodec, libswscale und libswresample für die Unterstützung des Exports von MP4-Sequenzen.
- wcslib für die Verarbeitung des Welt-Koordinatensystems (WCS), Objektnamen und photometrische Farbkalibrierung.
- Exiv2 um Metadaten der Bilder zu verwalten.

- libcurl für Astrometrie und photometrische Anfragen.
- libgit2 um einen lokalen Klon des Siril-Scripts-Repositorys zu verwalten.

Build-Abhängigkeiten

Für die Installation aus dem Quellcode müssen Sie die Basisentwicklungspakete installieren:

```
git, autoconf, automake, libtool, intltool, pkg-tools, make, cmake, gcc, g++
```

Die Compiler gcc und g++ aus dieser Liste können durch clang und clang++ ersetzt werden (wir verwenden sie für die Entwicklung), wahrscheinlich auch durch andere.

The autotools packages (autoconf, automake, probably some others) can be replaced by meson.

Grundsätzlicher Build-Prozess

Siril kann entweder mit autotools oder mit meson kompliert werden.

Meson

Der empfohlene Weg ist es, meson und ninja zu verwenden:

```
meson setup _build --buildtype release

cd _build
ninja
ninja install
```

Um einige Abhängigkeiten oder Funktionen zu deaktivieren, verwenden Sie je nach Fall die Meson-Optionen „-Dfeature=false“ oder „-Denable-feature=yes“.

Die folgende Tabelle listet alle konfigurierbaren Optionen auf.

Option	Typ	Wert	Auswahl	Beschreibung
relocatable-bundle	combo	platform-default	['yes', 'no', 'platform-default']	bauen mit Ressourcen, die unter demselben Präfix gebündelt werden
openmp	Boolean	true	Unbekannt	bauen mit OpenMP-Support
exiv2	Boolean	true	Unbekannt	bauen mit exiv2-Support
libgit2	Boolean	true	Unbekannt	bauen mit libgit2 Git-Integration
libraw	Boolean	true	Unbekannt	bauen mit LibRaw-Support
libtiff	Boolean	true	Unbekannt	bauen mit TIFF-Support
libjpeg	Boolean	true	Unbekannt	bauen mit JPEG-Support
libjxl	Boolean	true	Unbekannt	bauen mit JPEG-XL-Support
libpng	Boolean	true	Unbekannt	bauen mit PNG-Support
libheif	Boolean	true	Unbekannt	bauen mit HEIF-Support
libXISF	Boolean	true	Unbekannt	bauen mit XISF-Support
ffms2	Boolean	true	Unbekannt	bauen mit FFMS2-Support
ffmpeg	Boolean	true	Unbekannt	Bauen mit FFmpeg-Support
libcurl	Boolean	true	Unbekannt	Benutze libcurl für Webzugriffe
criterion	Boolean	false	Unbekannt	bauen mit Criterion-Support
flatpak-id	Zeichenkette		Unbekannt	set by flatpak-builder if building a flatpak

Autotools

Die Autotools-Methode ist in der Unix-Welt bekannt. Sobald der Quellcode heruntergeladen und die Voraussetzungen installiert wurden, ist die allgemeine Methode zum Erstellen wie folgt:

```
./autogen.sh
make
make install
```

Für die letzte Zeile benötigen Sie möglicherweise Superuser-Rechte.

Es kann sein, dass Sie dem Compiler bestimmte Optionen übergeben möchten, z.B. wenn Sie die Optimierung und die Installation in `/opt` anstelle des Standardverzeichnisses `/usr/local` wünschen:

```
CFLAGS='-mtune=native -O3' ./autogen.sh --prefix=/opt
```

Um Siril zu starten, lautet der Befehlsname **`siril`** oder **`siril-cli`**.

Installation auf Debian-ähnlichen Systemen

Es kann sein, dass Sie ein `.deb`-Paket bauen möchten, anstatt eine nicht-paketierte Version zu benutzen. In diesem Fall lesen Sie diese Hilfe. Insbesondere, um die Abhängigkeiten zu installieren, können Sie diesen Befehl verwenden:

```
apt build-dep siril
```

Ansonsten finden Sie hier die Liste der Pakete für die aktuelle Version:

- Für das Build-System benötigte Pakete:

```
autoconf automake make gcc g++ libtool intltool pkg-config cmake
```

- Liste der Pakete für obligatorische Abhängigkeiten:

```
libgtk-3-dev libcfitsio-dev libfftw3-dev libgsl-dev libopencv-dev
liblcms2-dev wcslib-dev libgtksourceview-4-dev
```

- Liste der Pakete für optionale Abhängigkeiten:

```
libcurl4-gnutls-dev libpng-dev libjpeg-dev libtiff5-dev
libraw-dev gnome-icon-theme libavformat-dev libavutil-dev libavcodec-dev
libswscale-dev libswresample-dev libgit2-dev libheif-dev
libexiv2-dev libjxl-dev liblz4-dev libpugixml-dev libxisf-dev
```

für die Unterstützung von Videodateien (AVI und andere):

```
libffms2-dev
```

Installation auf Arch Linux

Zwei Pakete sind im AUR verfügbar: **`siril`** und `siril-git`. Laden Sie `PKGBUILD` oder das Repository herunter, installieren Sie die Abhängigkeiten, lassen Sie makepkg laufen, um das Paket zu bauen und `pacman -U`, um es zu installieren.

Abhängigkeiten (obligatorisch und einige optional):

```
pacman -S base-devel cmake git intltool gtk3 fftw cfitsio gsl opencv
exiv2 libraw wcslib gtksourceview4
```

LittleCMS version

Siril benötigt zum Erstellen lcms2 >= 2.14. Diese Version ist erforderlich, um eine Optimierung des Farbmanagementcodes zu ermöglichen. Eine ausreichende Version ist in Debian Testing und Ubuntu 23.04 verfügbar, die aktuelle Ubuntu LTS-Version bietet jedoch eine ältere Version und andere Betriebssystemdistributionen möglicherweise auch. Wenn Sie ein älteres Betriebssystem verwenden, das lcms2 >= 2.14 nicht bereitstellt, können Sie Pakete entweder manuell von einer neueren Version Ihrer Distribution installieren oder lcms2 aus dem Quellcode erstellen und installieren. Die lcms2-Quelle ist in „ihrem Git-Repository <https://github.com/mm2/Little-CMS>“ verfügbar.

Fehler beim Erstellen

Jeder Commit für Siril Git wird automatisch in einer Standard-Build-Umgebung für Linux, Windows und MacOS mithilfe der Gitlab-CI-Infrastruktur erstellt. Dies bedeutet, dass wir sehr zuversichtlich sind, dass der Master-Zweig sowie getaggte Releases **erfolgreich** erstellt werden, sofern eine korrekt eingerichtete Build-Umgebung mit den erforderlichen installierten Abhängigkeiten vorhanden ist.

Wenn ein Build-Fehler auftritt, deutet dies wahrscheinlich auf ein Problem mit Ihrer Build-Umgebung oder falsch installierte Abhängigkeiten hin. Denken Sie daran, dass viele Distributionen eine separate Installation von Entwicklungspaketen erfordern, die die erforderlichen Header-Dateien enthalten. Überprüfen Sie den CI-Bericht für den Git-Commit, den Sie erstellen möchten. Für den unwahrscheinlichen Fall, dass ein Build-Fehler angezeigt wird, können Sie sicher sein, dass das Team daran arbeitet, das Problem zu beheben. Andernfalls, wenn die CI-Pipeline grüne Häkchen anzeigt, müssen Sie alle Probleme mit Ihrer eigenen Build-Umgebung überprüfen und beheben.

Wenn Sie immer noch der Meinung sind, dass Sie ein Build-Problem gefunden haben, das von der CI-Pipeline nicht gemeldet wurde – beispielsweise wenn Sie auf einer anderen Plattform wie BSD bauen, die die Entwickler nicht regelmäßig verwenden –, können Sie gerne ein Problem melden auf gitlab.

Beachten Sie, dass Probleme nur für den Master-Zweig oder getaggte Releases gemeldet werden sollten. Wenn Sie neue Funktionen in Merge-Requests testen, geben Sie bitte Feedback in den Kommentaren zum entsprechenden Merge-Request.

System Python

If you are building Siril yourself you need to ensure you have a working system Python installation including pip and venv. Note the last point because on some systems (at least Debian / Ubuntu based systems, possibly others) Python is split into many packages and you must ensure the packages that provide pip and venv are installed.

KAPITEL 2

Grafische Benutzeroberfläche

Die grafische Benutzeroberfläche (GUI) ermöglicht es Ihnen, Ihre Bilder manuell zu bearbeiten, aber auch Skripte zu verwenden oder Befehle einzugeben. Um zu erfahren, sie Sie Siril im Headless-Modus verwenden, lesen Sie bitte diesen Abschnitt: *Headless Modus*.

Die Benutzeroberfläche von Siril wurde mit GTK, einem freien und quelloffenem und plattformübergreifendem Toolkit für die Erstellung von Benutzeroberflächen geschrieben. Derzeit wird Version 3 verwendet.

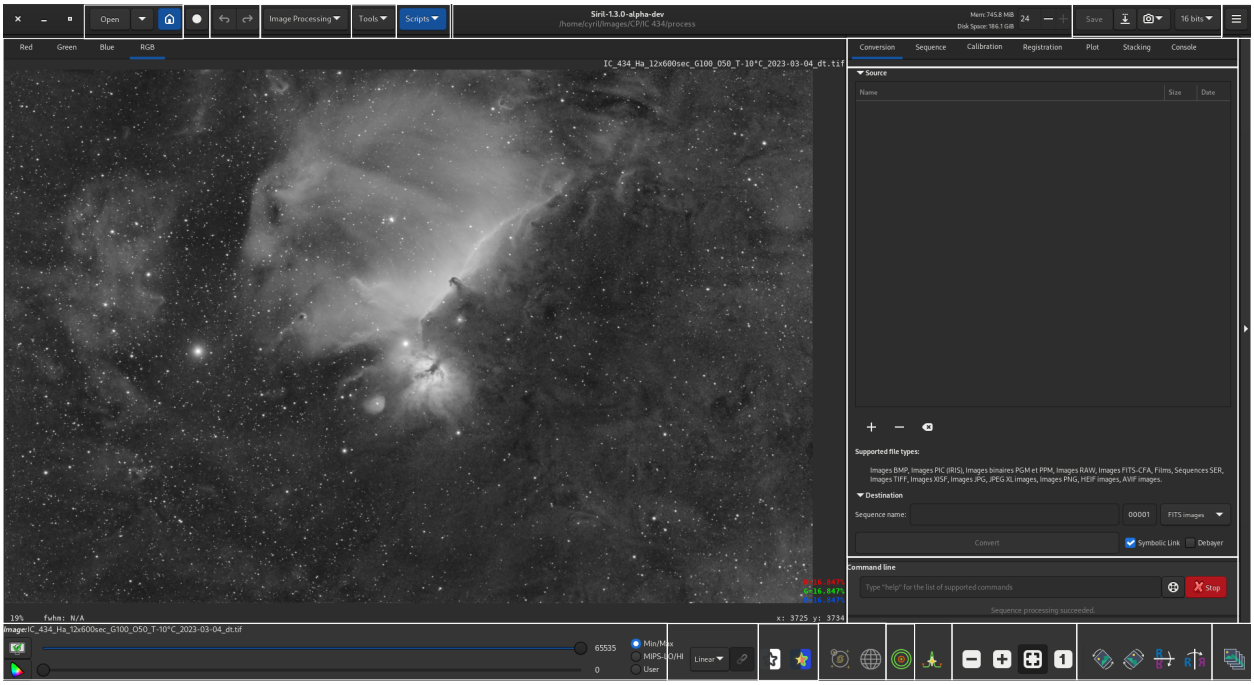
Die folgenden Unterabschnitte führen Sie durch das Hauptfenster der Benutzeroberfläche und die nützlichen Menüs.

2.1 Haupt-Benutzerinterface

Beim Start von Siril wird das Haupt-Benutzerinterface geöffnet.

Bemerkung

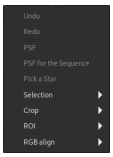
Klicken sie irgendwo auf das Bild unten, um die Funktionen anzuzeigen.



Bildbereich

Dieser Bereich zeigt das aktuell geladene Bild an. Klicken sie auf *Rot*, *Grün* oder *Blau* um zwischen verschiedenen Ebenen umzuschalten (nur bei Farbbildern, bei Schwarzweißbildern gibt es nur eine einzige Registerkarte *B/W*).

Klicken Sie mit der rechten Maustaste auf das Bild, um ein Kontextmenü anzuzeigen:



> **Tipp**
>
> Falls kein Bild geladen ist, öffnet ein Doppelklick auf den Bildbereich den Dialog *Open*.

2.1.1 Wiederherstellen

Macht die letzte Operation rückgängig.

2.1.2 Wiederholen

Macht das letzte Undo rückgängig.

2.1.3 PSF

Berechnet eine PSF für die aktuelle Auswahl und führt eine schnelle Photometrie durch.

2.1.4 PSF für die Sequenz

Berechnet eine PSF für die aktuelle Auswahl für alle Bilder der aktuellen Sequenz.

2.1.5 Stern wählen

Wendet die Sternerkennung auf die aktuelle Auswahl an. Das Ergebnis wird im Dialog :guilabel:`Dynamische PSF`geöffnet. Beachten Sie, dass dieser Vorgang nachsichtiger ist als die Vollbild-Sternsucher-Routine, da er davon ausgeht, dass Sie eine Auswahl getroffen haben, die einen Stern enthält: Er kann daher verwendet werden, um Sterne zu erkennen, die von der Vollbild-Sternsucher-Funktion übersehen wurden.

2.1.6 Auswahl

Wendet Beschränkungen auf den Auswahlbereich an.

2.1.7 Beschneiden

Stellt ein Interface zu den Funktionen *Crop* und *Crop & rotate* bereit.

> **Tipp**
>
> If a CFA image is loaded, the crop will be constrained to a CFA boundary so that the effective CFA pattern is unchanged for future operations. For Bayer CFA patterns this means the crop selection will snap to a multiple of 2x2 and may shift by up to 1 pixel in x and y directions in order to align with the Bayer pattern start. For X-Trans patters the crop selection will snap to a multiple of 6x6 and may shift by up to 5 pixels in each direction.

> **Warnung**
>
> Rotating a CFA image by arbitrary angles will destroy the CFA pattern: you should progress your workflow to the point where the CFA pattern has been debayered or Bayer drizzled before doing anything that applies rotation or scaling.

2.1.8 ROI

Bietet eine Schnittstelle zum Festlegen oder Löschen der "Region of Interest" für die ROI-Verarbeitung. Hinweis: Dieser Menüeintrag ist nicht verfügbar, wenn der ROI-Modus in der Registerkarte "Benutzeroberfläche" auf "Automatisch aus Auswahl" eingestellt ist.

2.1.9 RGB-Ausrichtung

Richtet die RGB-Kanäle des Bildes aus. Dies unterstützt eine Reihe von Registrierungsmethoden für die Ausrichtung:

- **Globale Sternausrichtung in 2 Durchgängen** Hier wird die globale Sternausrichtungsroutine in 2 Durchgängen mit COG(Schwerpunkt)-Rahmen verwendet (damit die Bildabmessungen erhalten bleiben). Dieser Algorithmus passt sich mit 8 Freiheitsgraden an, so dass er Bildverschiebung, Drehung und Verzerrung korrigieren kann. Dies ist der empfohlene Algorithmus für Bilder, die Sterne enthalten. Dieser Algorithmus ist global und erfordert keine Auswahl.
- **KOMBAT-Ausrichtung** Hier wird die KOMBAT-Ausrichtungsmethode verwendet. Sie ist in erster Linie für die Ausrichtung von Planeten gedacht, kann aber auch für Deep Space-Bilder verwendet werden. Er bietet nur eine Verschiebungsregistrierung, ist aber schnell. Bei diesem Algorithmus muss eine Auswahl getroffen werden.
- **Ein-Stern-Registrierung** Dies ist ein älterer Algorithmus, der eine reine Verschiebungs-Registrierung auf der Grundlage der Bewegung eines einzelnen Sterns ermöglicht. Dies ist sehr schnell, aber die Registrierung kann nicht so viele Freiheitsgrade bewältigen wie die globale 2-Pass-Sternausrichtung. Er kann geeignet sein, wenn Sie sicher sind, dass Sie nur die Bildverschiebung korrigieren müssen. Bei diesem Algorithmus muss ein einzelner Stern ausgewählt werden.
- **Bildmusterausrichtung** Bei dieser Methode wird die Ausrichtung anhand der Fourier-Transformationen der Bilder vorgenommen. Sie eignet sich für Deep-Sky- und Planetenbilder, bietet jedoch nur eine Shift-Registrierung und ist recht langsam. Dieser Algorithmus verwendet eine quadratische Auswahl: Es muss eine Auswahl getroffen werden, und nicht quadratische Auswahlen werden so angepasst, dass sie quadratisch sind.

> **Warnung**
>
> Das Werkzeug ist **RGB**-Ausrichtung. Es ist nicht für das Ausrichten von nicht ausgerichteten **LRGB**-Kompositionen gedacht und funktioniert auch nicht damit. Dazu müssen Sie entweder die zu komponierenden Ebenen vor der Komposition ausrichten oder die Ausrichtung mit dem RGB-Ausrichtungswerkzeug vornehmen.

Zurück zum Bild

Öffnen

Klicken Sie auf diese Icons (von links nach rechts), um:

- Eine Datei zu öffnen
- Eine kürzlich verwendete Datei zu öffnen
- *Arbeitsverzeichnis* wechseln

Zurück zum Bild

Live-Stacking

Klicken Sie auf diese Schaltfläche, um eine *Live-Stacking*-Sitzung zu starten.

Zurück zum Bild

Rückkgängig machen/Wiederherstellen

Benutzen Sie diese Schaltflächen, um die letzten Aktionen rückgängig zu machen/wiederherzustellen. Dies ist nur möglich, wenn die letzte Aktion über die grafische Benutzeroberfläche und nicht durch Eingabe eines Befehls durchgeführt wurde.

Zurück zum Bild

Bildverarbeitung

Klicken sie auf diese Schaltfläche, um das Menü *Verarbeitung* anzuzeigen.

Zurück zum Bild

Werkzeuge

Klicken sie auf diese Schaltfläche, um das Menü *Werkzeuge* anzuzeigen.

Zurück zum Bild

Skripte

Klicken Sie auf diese Schaltfläche um die *Skripte* anzuzeigen und zu starten.

Zurück zum Bild

Informationsleiste

Diese Leiste zeigt die aktuelle Version von Siril und den Pfad zum aktuellen *Arbeitsverzeichnis*.an.

- Auf der rechten Seite werden auch Informationen über den verfügbaren Arbeits- und Festplattenspeicher angezeigt.
- Sie können die maximale Anzahl der von Siril verwendeten Prozessorkerne mit den Schaltflächen + und - ändern.

Zurück zum Bild

Speichern

Diese Schaltflächen werden benutzt um Ihre Ergebnisse zu speichern:

- Speichert (überschreibt) das aktuelle Bild.
- Speichert unter einem anderen Dateinamen und/oder Dateierweiterung.

  In der Dropdown-Liste unten rechts können Sie den Typ des zu speichernden Bildes auswählen. Die Erweiterung wird automatisch zum Dateinamen hinzugefügt. Wenn Sie jedoch im Modus *Unterstützte Bild-Formate* bleiben, können Sie jede von Siril untersützte Dateierweiterung manuell hinzufügen, und die Datei wird im korrekten Format gespeichert.
- Erstellt einen Schnappschuss der aktuellen Ansicht (so wie sie auf dem Bildschirm zu sehen ist, d.h. die Vorschau wird, falls aktiviert, gestreckt). Es gibt zwei mögliche Optionen: Entweder wird der Schnappschuss in der Zwischenablage gespeichert, oder er wird direkt auf die Festplatte in das *Arbeitsverzeichnis* kopiert.
- Ändert die Bittiefe des aktuellen Bildes. Sie haben die Wahl zwischen 16-Bit und 32-Bit.

Zurück zum Bild

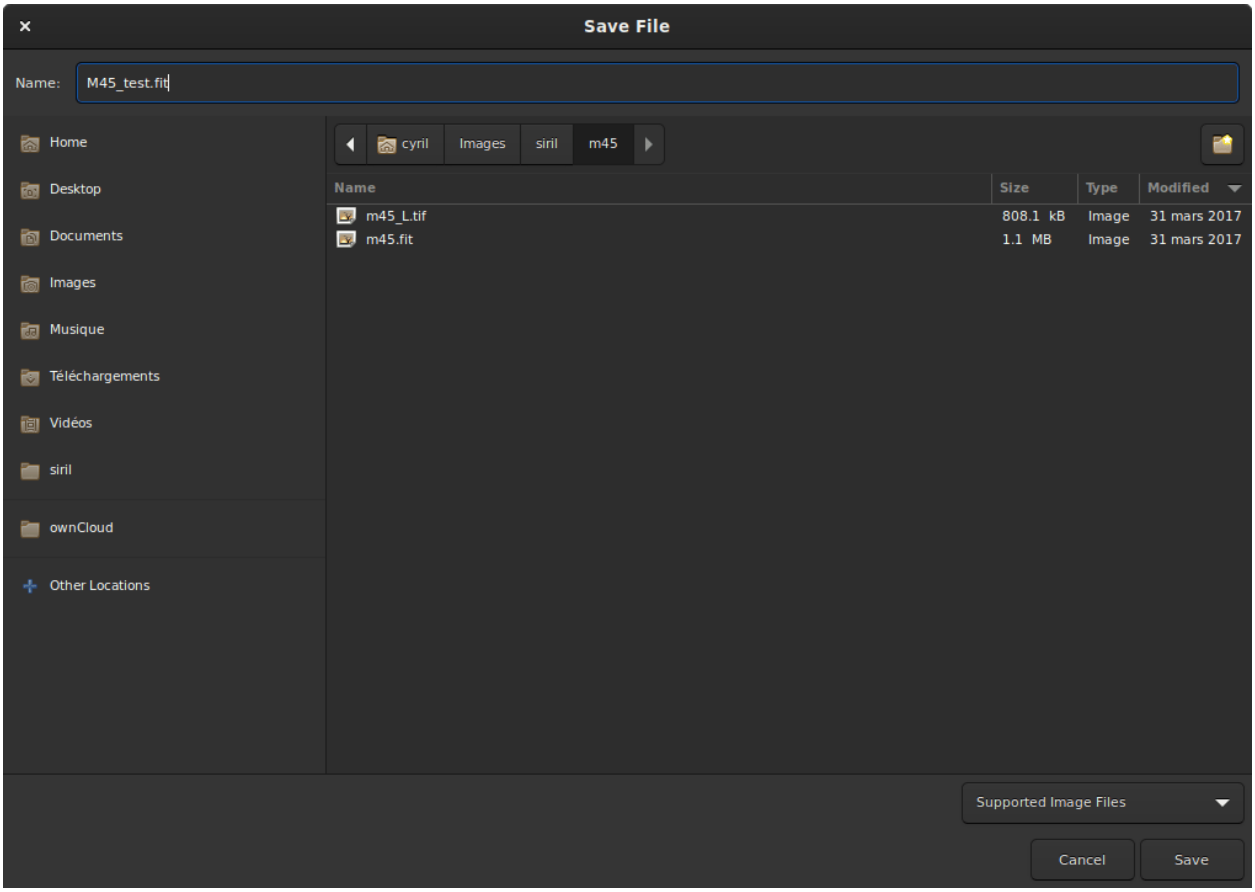


Abb. 1: Dialogbox "Speichern".

Hamburger-Menü

Öffnet das Hauptmenü, auch *Burgermenü* genannt. Ermöglicht den Zugriff auf *Einstellungen*, die Dokumentation und vieles mehr.

Zurück zum Bild

Tabs/Registerkarten

Wählt eine der Registerkarten. Sie können auch mit den Tastenkombinationen `F1` bis `F7` zwischen den verschiedenen Registerkarten wechseln.

Weitere Einzelheiten sind hier zu finden:

Tabs/Registerkarten	Schlüssel
Tab *Umrechnung*	`F1`
Tab *Sequenz*	`F2`
Tab *Kalibrierung*	`F3`
Tab *Registrierung*	`F4`
Tab *Grafischer Plot*	`F5`
Tab *Stacking*	`F6`
Konsole	`F7`

Zurück zum Bild

Registerkartenfenster

Zeigt die Besonderheiten der aktuell ausgewählten Registerkarte an.

Zurück zum Bild

Verhalten der Maus

Siril verwendet GTK-Zuordnungen für die primäre, sekundäre und mittlere Maustaste. Diese sind normalerweise den physischen Maustasten links, rechts und Mitte (`Scroll`-Radtaste) zugeordnet, können jedoch in bestimmten Konfigurationen anders zugeordnet sein (z. B. können Linkshänder die linke und rechte Taste vertauscht haben).

1. **Haupt-Maustaste**

   Die Haupt-Maustaste wird für eine Reihe von Zwecken verwendet:

   - Um Bereiche eines Bildes auszuwählen
   - Wenn `Strg` gedrückt wird (oder `Cmd` auf MacOS), um das Bild zu verschieben
   - Wenn das Intensitätsprofilierungswerkzeug aktiv ist, um eine Linie zu ziehen, entlang der das Intensitätsprofil gezeichnet werden soll, und um Punkte auszuwählen, die bekannten Wellenlängen / Wellenzahlen in der Spektroskopieeinstellung entsprechen.
   - Um im RGB-Kompositionswerkzeug den Rotationsmittelpunkt bei manueller Ausrichtung auszuwählen
   - Im Werkzeug zur Gradientenentfernung um Probepunkte zu zeichnen
   - Im Photometrie-Modus um Photometrie an einem Stern durchzuführen
   - Um die Positionen für die Registrierungsvorschau auszuwählen
   - Ein Doppelklick mit der Haupt-Maustaste öffnet den Dialog *Öffnen* wenn kein Bild geladen ist.

2. ** Zweite Maustaster **

   Die zweite Maustaste wird für eine Reihe von Zwecken verwendet:

   - Um das Bild-Kontextmenü zu öffnen (außer im Photometrie-Modus)
   - Im Werkzeug zur Gradientenentfernung um Probepunkte zu löschen
   - Im Photometrie-Modus um Photometrie an einem ausgewählten Stern für alle Bilder der aktuell geladenen Sequenz durchzuführen (wenn keine Sequenz geladen ist hat das keinen Effekt)

3. **Mittlere Maustaste**

   Die mittlere Maustaste wird für die Folgenden Zwecke verwendet:

   - Um das Bild in der aktuellen Zoomstufe zu verschieben
   - Wenn `Strg` gedrückt ist (oder `Cmd` auf MacOS), um eine quadratische Auswahl mit passender Größe für die Photometrie zu erstellen
   - Doppelklick justiert die Zoomstufe auf einen vordefinierten Wert. Dieses Verhalten kann in den *Einstellungen* im Tab *Benutzeroberfläche* konfiguriert werden. Die Wahlmöglichkeiten sind:
     - Immer passend zoomen
     - Immer auf 100% zoomen, zentriert am Mauscursor
     - Zwischen passenden Zoom und 100% Zoom am Mauscursor wechseln

4. **Scrollrad**

   - Das :kbd:`Scroll`rad wird benutzt um die Zoomstufe einzustellen

Kommandozeile

Geben Sie einen *command* ein und drücken Sie `Enter`.

- Sie können die Schaltfläche am Ende der Zeile anklicken, um Hilfe bei der Benutzung zu erhalten.
- Sie können den gerade ausgeführten Prozess auch abbrechen, indem Sie auf die Schaltfläche *Stop* klicken.

Zurück zum Bild

Erweitern

Klicken Sie auf diese Leiste, um den gesamten Bereich der Registerkarte/des Registerkartenfenstes zu erweitern oder zu verkleinern.

Zurück zum Bild

Bildlaufleisten

Verwenden Sie den oberen und unteren Schieberegler, um den Weiß- und Schwarzpunkt des Vorschaubildes anzupassen (im Modus Linear).

> **Tipp**
>
> Klicken Sie auf den Namen des geladenen *Bildes* oder der geladenen *Sequenz*, um den Namen in die Zwischenablage zu kopieren (nützlich zum Einfügen in einen Befehl).

Zurück zum Bild

Vorschau-Modus

Wählen Sie den Vorschaumodus für das geladene Bild aus den folgenden Möglichkeiten:

- Linear
- Logarithmisch
- Quadratwurzel
- Quadratisch
- Asinh
- Autostretch (aktivieren Sie die Checkbox "Hohe Auflösung" um eine tiefere bis zu 24-Bit LUT statt der standardmäßigen 16-Bit-Tabelle zu verwenden)
- Histogramm

Im Autostretch-Modus mit Farbbildern, aktiviert/deaktiviert diese Checkbox auf der rechten Seite die Kanalverknüpfung. Wenn die Verknüpfung aufgehoben ist, werden die 3 Farbebenen unabhängig voneinander gestreckt, um ein farblich neutraleres Bild zu erhalten.

> **Warnung**
>
> Hierbei handelt es sich nur um eine Vorschau des Bildes, nicht um die tatsächlichen Daten (außer bei Auswahl des linearen Modus). Vergessen Sie nicht, Ihre Bilder zu strecken, bevor Sie sie speichern.

Zurück zum Bild

Spezielle Ansichten

Verwenden Sie diese Schalter um Vorschaubilder anzuzeigen:

- mit invertierten Farben
- in Falschfarben

Zurück zum Bild

Astrometrische Werkzeuge

Benutzen Sie diese Schalter um Folgendes anzuzeigen:

- *Astrometrische Beschriftungen*
- Koordinaten-Gitternetz

> **Warnung**
>
> Es muss eine astrometische Lösung für das Bild gefunden worden sein, damit diese Schaltflächen aktiv sind.

Zurück zum Bild

Schnelle Photometrie

Benutzen Sie diesen Schalter um den Modus *Schnelle Photometrie* zu aktivieren.

Zurück zum Bild

Intensitätsprofil

Benutzen Sie diesen Schalter um den Modus *Intensitätsprofil* zu aktivieren.

Zurück zum Bild

Zoom

Benutzen Sie diese Schaltflächen um:

- Herauszoomen
- Hereinzoomen
- Zoom an die verfügbare Fenstergröße anpassen
- Zoom auf die tatsächliche Größe

> **Tipp**
>
> `Strg+linke Maustaste` ermöglicht die Navigation im Bild

> **Tipp**
>
> `Strg+Mausrad` vergrößert/verkleinert und `Strg + 0 / 1` vergrößert auf die Fenstergröße bzw. 100%.

> **Tipp**
>
> `Strg+Umschalt` und Ziehen mit der primären Maustaste misst den Abstand zwischen zwei Punkten. Wenn genügend Metadaten vorhanden sind, wird die Messung in Grad, Minuten und Bogensekunden angegeben, ansonsten in Pixeln.

Zurück zum Bild

Geometrische Transformationen

Benutzen Sie diese Schaltflächen um:

- Nach links drehen
- Nach rechts drehen
- Entlang der horizontalen Achse zu spiegeln
- Entlang der vertikalen Achse zu spiegeln

Zurück zum Bild

Bildauswahl

Klicken Sie auf diese Schaltfläche um die *Bildauswahl* zu öffnen.

Zurück zum Bild

2.2 Main Toolbar

The main toolbar is located at the bottom of Siril's main window. It provides quick access to display controls, analysis tools and basic image transformation commands.

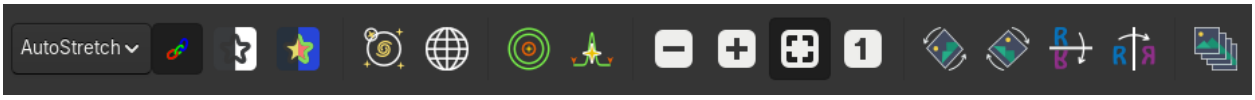


Abb. 2: The Siril main toolbar.

> **Bemerkung**
>
> The toolbar is only active when an image is loaded in memory.

2.2.1 Display Group

Display mode

The first control is a drop-down menu button showing the current display mode (`Linear` by default). Clicking it opens the list of all visual modes available in Siril:

- **Linear** — direct rendering of pixel values. This is the reference mode.
- **Logarithm** — applies a logarithmic transfer function, compressing bright areas and bringing out faint details.
- **Square root** — applies a square root transfer function, a gentler alternative to the logarithm.
- **Squared** — applies a quadratic transfer function, boosting bright areas at the expense of the shadows.
- **Asinh** — applies an inverse hyperbolic sine transfer function, offering a smooth and natural-looking stretch.
- **AutoStretch** — automatically computes and applies a stretch for a visually pleasing result. A **High definition** sub-option is available to increase the precision of the stretch computation.
- **Histogram** — applies a stretch based on the image histogram for a balanced display.

> **Warnung**
>
> Only **Linear** mode (with both visualisation sliders at their minimum and maximum positions) reflects the true pixel values. All other modes are purely visual aids. Stretch the histogram before exporting to an external tool.

Channel linking

Immediately to the right of the mode selector is a toggle button showing a chain link icon.

- **Closed chain** — the R, G and B channels are linked in AutoStretch mode.
- **Broken chain** — the channels are stretched independently.

2.2.2 Visual Rendering Group

Negative view

Toggles the display between the normal view and the **negative view** of the image. Useful for detecting subtle artefacts or faint gradients.

False colour / rainbow colormap

Toggles between normal display and a **false colour rendering** (rainbow palette). This mode helps visualise intensity variations across the image more clearly.

2.2.3 Astrometry Group

Object annotation

- **Left-click** — displays the names of celestial objects if WCS (World Coordinate System) information is available in the image.
- **Right-click** — opens the list of available astrometric catalogues to select the annotation source.

Celestial grid

Shows or hides the **celestial coordinate grid** (right ascension / declination) if WCS information is available in the image.

2.2.4 Analysis Group

Photometry / PSF

Switches to *photometry / PSF mode*. In this mode:

- A click on a star performs a PSF (Point Spread Function) fit and displays the photometric parameters.
- If a sequence is loaded, a **right-click** on the displayed image applies the PSF/photometry analysis to the entire sequence.

Intensity profile cut

Activates the *intensity profile cut tool*: draw a segment between two points on the image to display a graph of the pixel values along that line.

2.2.5 Zoom Group

Herauszoomen

Reduces the zoom factor (shrinks the displayed image). Usual keyboard shortcut: `-`.

Hereinzoomen

Increases the zoom factor (enlarges the displayed image). Usual keyboard shortcut: +.

Fit to window

Toggle button. When active, the image is **automatically scaled** to fill the available area of the window.

Actual size — 1:1

Displays the image at its **actual size** (1 image pixel = 1 screen pixel).

2.2.6 Geometric Transformations Group

Rotate 90° counter-clockwise

Applies a **90° counter-clockwise rotation** to the image.

Rotate 90° clockwise

Applies a **90° clockwise rotation** to the image.

Horizontal mirror

Applies a **horizontal flip** (reflection about the vertical axis).

Vertical mirror

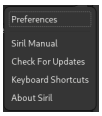
Applies a **vertical flip** (reflection about the horizontal axis).

2.2.7 Sequence Group

Sequence image list

Shows or hides the **side panel listing the images in the current sequence**, along with the registration data associated with each frame.

2.3 Hamburger-Menü



Voreinstellungen

Öffnet das Menü *Einstellungen*.

Siril-Handbuch

Öffnet die Online-Dokumentation <https://siril.readthedocs.io>`_.

Nach Updates suchen

Prüft, ob eine neuere Version verfügbar ist.

Tastenkürzel

Öffnet ein Fenster, das alle verfügbaren *Tastenkürzel* anzeigt.

Über Siril

Öffnet einen Dialog mit Versionsinformationen und Credits.

2.4 Werkzeugmenü

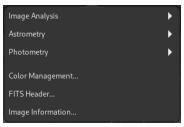


Abb. 3: Werkzeugmenü

In diesem Menü, das mit der Version 1.4 eingeführt wurde, sind die verschiedenen Werkzeuge von Siril zusammengefasst, wie z. B. Bildanalyse, Astrometrie, Photometrie usw.

Es besteht aus mehreren mehrstufigen Menüs und Untermenüs, die wie folgt aufgebaut sind:

- **Bildanalyse**
  - *Statistik*
  - *Hintergrundrauschen*
  - *Aberrationsprüfung*
  - *Verkippung anzeigen*
  - *Verzerrungen anzeigen*
  - *Dynamische PSF*
- **Astrometrie**
  - *Astrometrische Lösung*
  - *Astrometrische Beschriftungen*
- **Photometrie**
  - ref:*Vergleichssternliste erstellen <photometry/lightcurves:Generating a list of comparison stars>*
  - ref:*Automatische Lichtkurve <photometry/lightcurves:Automated Light Curve>*
  - **Ausgabe erzeugen**
    - *Lichtkurve (ETD-Format)*
    - *AAVSO Dateiformat*
- *Farbmanagement*
- *FITS Header*
- *Bildinformationen*

Detaillierte Informationen zu den einzelnen Punkten erhalten Sie, wenn Sie auf den jeweiligen Eintrag klicken.

2.5 Bildbearbeitungsmenü

Dieses Menü wird detailliert im Abschnitt *page* erklärt.

2.6 Verarbeitung von Teilbereichen

Siril bietet für einige Funktionen die Verarbeitung von Teilbereichen (ROI) an. Die ROI-Verarbeitung dient zwei Zwecken:

- Bei langsam zu berechnenden Bildoperationen wie der Dekonvolution kann nur ein kleiner Teil des Bildes bearbeitet werden, während die idealen Parameter experimentell ermittelt werden. Dies bedeutet, dass der Vorgang auf der kleinen ROI viel schneller abgeschlossen wird und die gewünschten Parameter viel schneller ausgewählt werden können.
- Bei Bildoperationen, die sich auf das Aussehen des Bildes auswirken, wie z. B. Stretching, Sättigungswerkzeuge usw., kann es sinnvoll sein, die ROI-Verarbeitung zu verwenden, um einen unbearbeiteten und einen bearbeiteten Bereich nebeneinander zu vergleichen.

Die ROI-Verarbeitung ist nicht für alle Bildoperationen in Siril implementiert. In der Tat gibt es viele Fälle, in denen es keinen Sinn macht, z. B. Geometrieoperationen, bei denen der Inhalt des ROI auf einen anderen Teil des Bildes abgebildet werden würde. Die ROI-Verarbeitung wird nicht für Bildoperationen unterstützt, die mehrere Eingabebilder benötigen, z. B. *Stern-Neuzusammensetzung*, *RGB-Zusammensetzung* oder *Pixelmath*.

2.6.1 Operationen mit Unterstützung der ROI-Verarbeitung

Die folgenden Bildoperationen unterstützen die ROI-Verarbeitung:

- **Deconvolution** Die Dekonvolution kann sehr langsam sein, daher ermöglicht die ROI-Verarbeitung ein schnelleres Ausprobieren von Optionen an einem kleinen Teil des Bildes, um die besten Einstellungen für Ihr Bild zu finden.
- **Rauschunterdrückung** Die Rauschunterdrückung kann auch langsam verarbeitet werden, so dass die ROI-Verarbeitung eine schnellere Erprobung der Optionen ermöglicht.
- **Asinh-Stretch** Ein Benutzer möchte vielleicht die Wirkung von Asinh-Stretch oder einer der folgenden Bildoperationen neben einem unbearbeiteten Bereich untersuchen. Die ROI-Verarbeitung ermöglicht dies.
- **Verallgemeinerte hyperbolische Streckung**
- **Histogramm Stretch**
- **Kurventransformation**
- **Farbsättigung**
- **Median-Filter**
- **Grün-Rauschen entfernen (SCNR)**
- **Kantenerhaltende Filter**
- **Unpurple-Filter (violette Säume entfernen)**

Keine anderen Bildoperationen unterstützen die ROI-Verarbeitung.

> **Bemerkung**
>
> Die ROI-Verarbeitung ist im Allgemeinen nicht dazu gedacht, nur einen Teil eines Bildes zu bearbeiten.
>
> Bei Bildoperationen mit automatisch aktualisierter Vorschau gilt die Vorschau für den ROI, aber wenn *Anwenden* angeklickt wird, wird das gesamte Bild aktualisiert.
>
> Bei Bildoperationen, die nicht über eine automatisch aktualisierte Vorschau verfügen, wird beim Festlegen eines ROI eine Schaltfläche *Vorschau im ROI* verfügbar. Diese Schaltfläche dient als manuelle Vorschau, mit der Sie eine Reihe von Einstellungen im ROI überprüfen können, bis Sie zufrieden sind. Sind die Einstellungen wie gewünscht, klicken Sie auf Anwenden, um den Vorgang auf das gesamte Bild anzuwenden.

2.6.2 Einstellung des ROI

Für die Einstellung des ROI gibt es zwei Möglichkeiten:

- **Manuell** Bei dieser Option muss die ROI manuell über das Menü der rechten Maustaste im Untermenü *ROI → Set ROI to selection* gesetzt und gelöscht werden.
- **Automatisch aus Auswahl** Diese Option synchronisiert die ROI automatisch mit der aktuellen Auswahl und löscht sie automatisch, wenn die Auswahl gelöscht wird.

Diese Optionen schließen sich gegenseitig aus und die bevorzugte Option kann in der Registerkarte *Benutzeroberfläche* des Dialogs *Einstellungen* konfiguriert werden.

2.6.3 Löschen des ROI

Im manuellen ROI-Einstellungsmodus befindet sich die Option zum Löschen des ROI im Menü der rechten Maustaste, im Untermenü *ROI → ROI Löschen*.

Im automatischen ROI-Einstellungsmodus löschen Sie einfach die aktuelle Auswahl und die ROI wird gelöscht.

Der ROI wird auch bei den folgenden Gelegenheiten automatisch gelöscht:

- Wenn das aktuelle Bild geschlossen wird,
- wenn ein neues Bild geöffnet wird,
- Wenn eine Sequenz geöffnet ist und das sichtbare Bild geändert wird.

2.6.4 Visualisierung

Wenn ein ROI gesetzt ist, wird er im Bildfenster als rot gestrichelter Rahmen im gleichen Stil wie das Auswahlfeld angezeigt. Wenn jedoch ein Dialogfeld für Bildoperationen geöffnet wird und die Bildoperation die ROI-Verarbeitung unterstützt (oder wenn eine ROI festgelegt wird, während ein solches Dialogfeld bereits geöffnet ist), ändert sich die Farbe des ROI-Umrisses in grün und zeigt damit an, dass die ROI-Verarbeitung für die aktuelle Operation aktiviert ist.

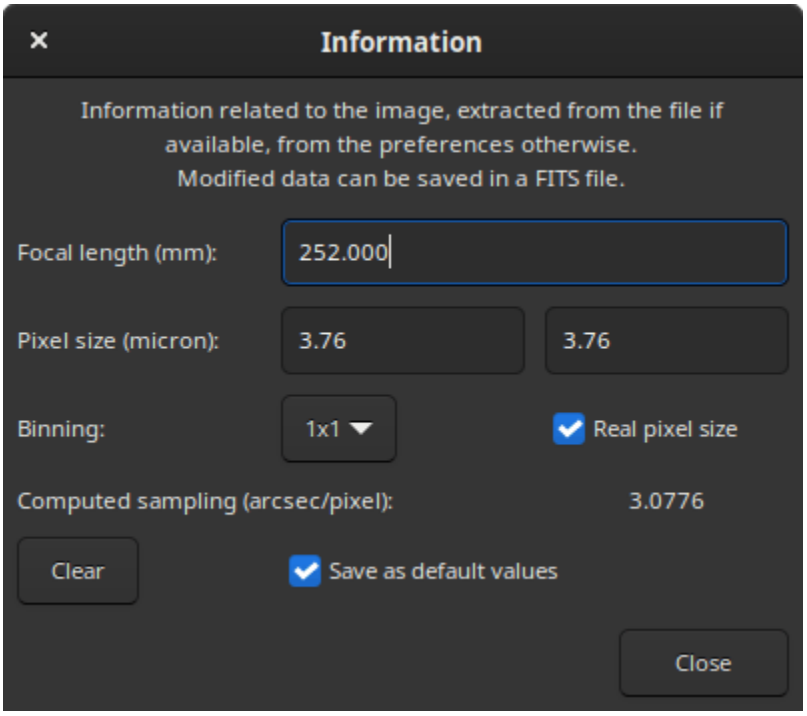
2.6.5 Siril Befehle

Siril-Befehle ignorieren die ROI, auch wenn sie im GUI-Modus von der Befehlszeile aus ausgeführt werden, und auch wenn dieselbe Bildoperation die ROI-Verarbeitung unterstützt, wenn sie von der GUI aus aufgerufen wird. Die ROI ist als schnelle Vorschau gedacht, während die Befehle das gesamte Bild bearbeiten sollen. Wenn eine ROI vorhanden ist, wenn ein Befehl ausgeführt wird, wird sie ignoriert, aber nicht gelöscht, so dass sie für nachfolgende Operationen, die die ROI-Verarbeitung unterstützen, weiterhin verfügbar ist.

2.6.6 Farbraumtransformationen

Jeder Vorgang, bei dem das ICC-Profil des Bildes entfernt, zugewiesen oder konvertiert wird, löscht die ROI, sofern eine solche festgelegt wurde.

2.7 Fenster mit Bildinformationen



Dieses Fenster enthält Informationen über die Abtastrate des geöffneten Bildes. Die Abtastrate, manchmal auch Auflösung oder Maßstab genannt, gibt an, wie viel Winkel des Himmels in einem Pixel zu sehen ist, wie er durch das Instrument gesehen wird. Sie hängt von zwei Dingen ab: von der Brennweite des Instruments und von der Pixelgröße des Sensors, die wiederum vom Binning-Modus abhängt.

FITS-Header können diese Informationen enthalten, wenn sie in der Erfassungssoftware eingestellt wurden. In diesem Fall sind dies die Werte, die in diesem Fenster angezeigt werden. Wenn die Metadaten des Bildes diese Informationen nicht enthalten, weil sie der Erfassungssoftware nicht bekannt sind oder weil das Dateiformat sie nicht unterstützt, ist dieses Dialogfeld dennoch verfügbar und wird mit *Standardwerten* gefüllt. Sie können geändert und für verschiedene Operationen von Siril verwendet werden, die Abtastinformationen erfordern, z. B. die Anzeige der FWHM in Bogensekunden anstelle von Pixeln.

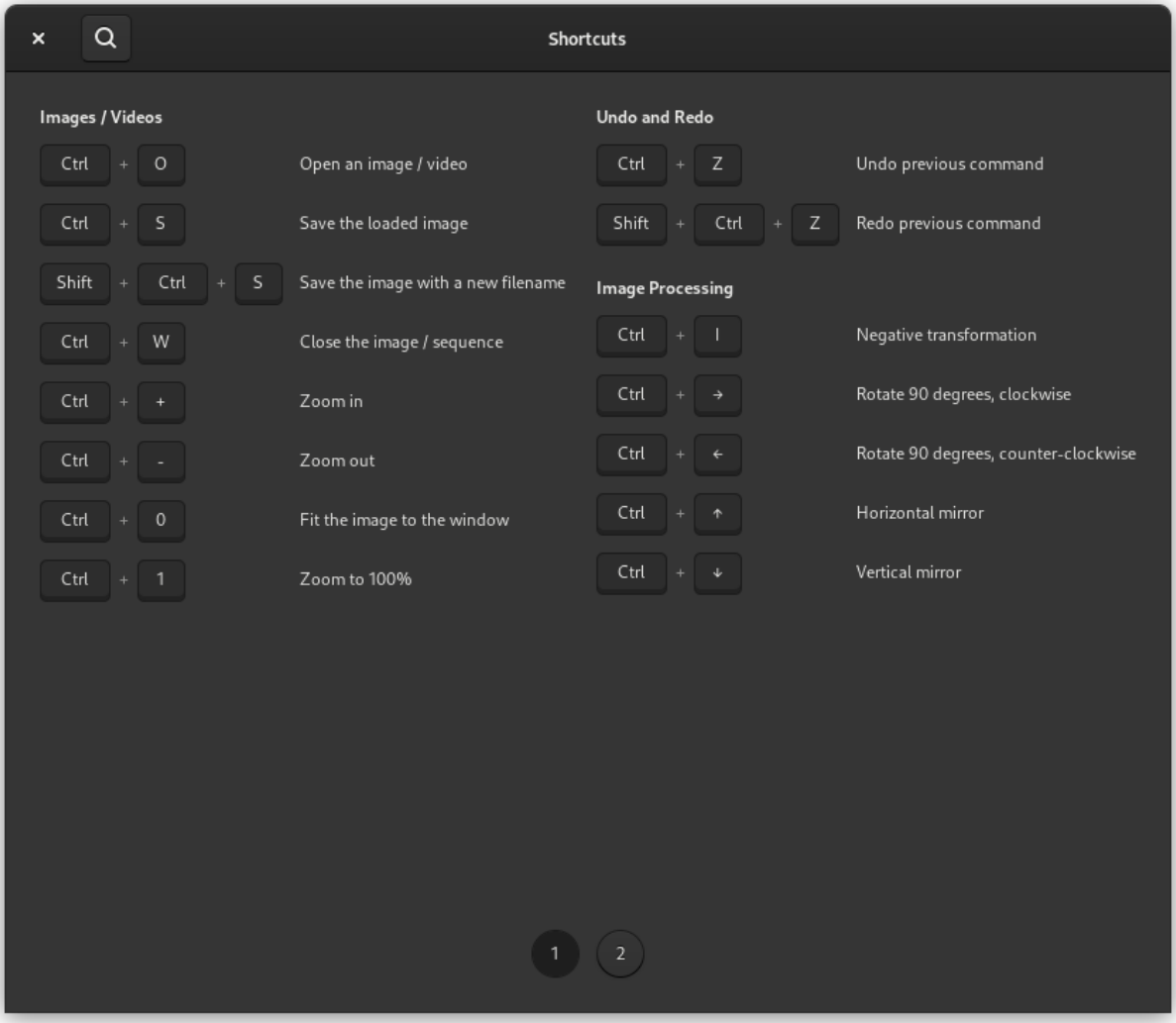
Standardwert ist "kein Binning" (1x1) und eine in den Einstellungen gespeicherte Brennweite und Pixelgröße. Die in den Einstellungen gespeicherten Werte können von diesem Dialog aus gesetzt werden, indem Sie die Schaltfläche *Als Standardwerte speichern* aktivieren, bevor Sie auf *Schließen* klicken. Sie können auch gesetzt werden, indem man eine astrometrische Auflösung des Bildes durchführt, auch *Plate Solve* genannt, wenn die Option, die Standardwerte zu aktualisieren, wenn ein Ergebnis gefunden wird, in den Voreinstellungen <preferences/preferences_gui:Astrometry>` aktiviert ist :ref:`.

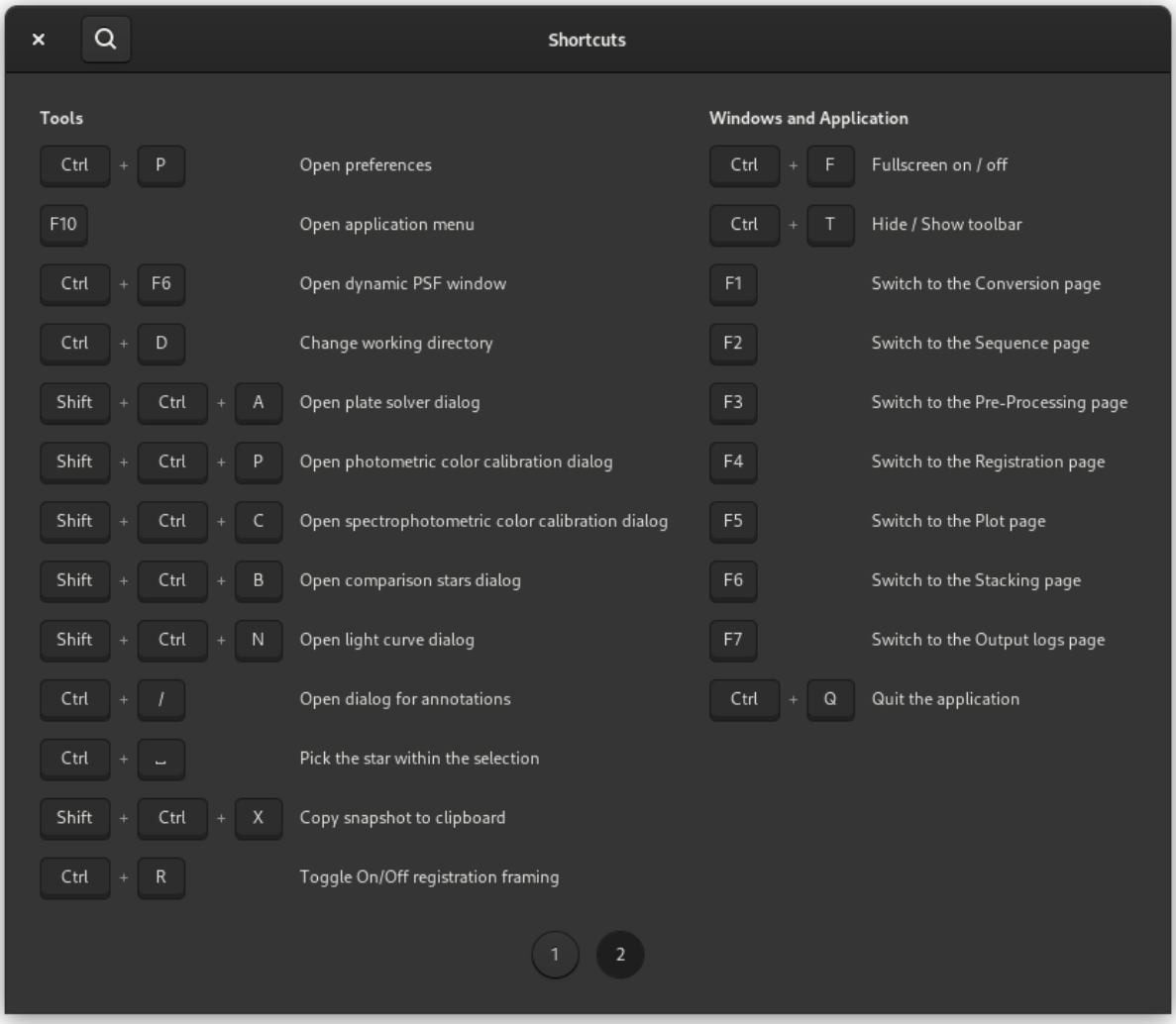
Die in diesem Fenster angezeigten Werte werden im aktuell geladenen Bild gespeichert, und wenn dieses Bild als FITS gespeichert wird, werden sie im FITS-Header gespeichert.

Die Binning-Verwaltung kann je nach Erfassungssoftware in zwei Formen erfolgen: Die tatsächliche Pixelgröße wird angegeben, muss aber mit dem Binning multipliziert werden (wenn *Echte Pixelgröße* angekreuzt ist), die bereits multiplizierte Pixelgröße wird angegeben (wenn nicht angekreuzt).

2.8 Tastenkürzel

Siril verwendet verschiedene Tastenkürzel für den Zugriff auf Bearbeitungswerkzeuge oder zur Bearbeitung des Programms und/oder der Bilder. Diese Tastenkürzel werden im Dialog *Tastenkürzel* aufgeführt, der über das Hamburgermenü ☰ zugänglich ist.




Shortcuts
Tools
Ctrl + P Open preferences
F10 Open application menu
Ctrl + F6 Open dynamic PSF window
Ctrl + D Change working directory
Shift + Ctrl + A Open plate solver dialog
Shift + Ctrl + P Open photometric color calibration dialog
Shift + Ctrl + C Open spectrophotometric color calibration dialog
Shift + Ctrl + B Open comparison stars dialog
Shift + Ctrl + N Open light curve dialog
Ctrl + / Open dialog for annotations
Ctrl + ␣ Pick the star within the selection
Shift + Ctrl + X Copy snapshot to clipboard
Ctrl + R Toggle On/Off registration framing
Windows and Application
Ctrl + F Fullscreen on / off
Ctrl + T Hide / Show toolbar
F1 Switch to the Conversion page
F2 Switch to the Sequence page
F3 Switch to the Pre-Processing page
F4 Switch to the Registration page
F5 Switch to the Plot page
F6 Switch to the Stacking page
F7 Switch to the Output logs page
Ctrl + Q Quit the application
1
2

KAPITEL 3

Arbeitsverzeichnis

Das Arbeitsverzeichnis (`wd`), auch als aktuelles Arbeitsverzeichnis (`cwd`) bekannt, ist das Verzeichnis, in dem Siril arbeitet. Seine Wahl ist ein entscheidender Schritt, insbesondere bei der Verwendung von Skripten. Die falsche Wahl von `cwd` ist für 90 % der *Skript*-Fehler verantwortlich. Dieser Ordner wird durch Klicken auf den *Home*-Button in Form eines Hauses ausgewählt: . Dies ist das Verzeichnis, in dem Siril standardmäßig Bilder speichert (sofern kein anderer Pfad angegeben ist) und auch das Verzeichnis, in dem nach *Sequenzen* gesucht wird.

Sobald das Verzeichnis ausgewählt wurde, kann sein Pfad einfach in der Titelleiste des Anwendungsfensters unter der Versionnummer überprüft werden, wie in der Abbildung unten dargestellt.

Abb. 1: Angezeigtes Arbeitsverzeichnis in der Titelleiste. Hier ist es ein Linux-Pfadname.

KAPITEL 4

Voreinstellungen

Voreinstellungen sind Einstellungen, die bei jeder Sitzung von Siril erhalten bleiben und die Ihre bevorzugten Auswahlmöglichkeiten für viele der Werkzeuge festlegen.

Seit Version 1.2.0 sind sie sowohl über die *Benutzeroberfläche* als auch programmatisch über die Befehle *set/get commands* zugänglich.

Standardmäßig findet sich die Voreinstellungsdatei hier:

- `~/.config/siril/configX.Y.ini` (Linux)
- `%LOCALAPPADATA%\siril\configX.Y.ini` (Windows)
- `~/Library/Application Support/org.free-astro.Siril/siril/configX.Y.ini` (MacOS)

Wobei *X* und *Y* die Haupt- und Nebenversionsnummern sind.

Falls Sie mehrere Konfigurationsdateien verwenden möchten, können Sie auswählen, mit welcher Sie beginnen möchten, indem Sie ein Terminal öffnen und folgenden Befehl eingeben:

```
siril -i path/to/my_other_config.ini
```

Warnung

Siril muss in Ihrem Suchpfad enthalten sein, um **`siril`** wie in der obigen Zeile verwenden zu können. Andernfalls verwenden Sie den vollständigen Pfadnamen zur Siril-Programmdatei.

4.1 Einstellungen (GUI)

Die Einstellungen sind über das Burgermenü ≡ oder über das Tastenkürzel `Strg` + `P` erreichbar. Es gibt 10 Seiten und jede Seite repräsentiert ein Thema. Die Einstellungen ermöglichen es mehr oder weniger fortgeschrittenen Benutzern, Siril so zu optimieren, dass es ihren Bedürfnissen am besten entspricht. Einige Einstellungen können sich negativ auf die Leistung von Siril auswirken, daher ist es ratsam, die Einstellungen nur zu verändern, wenn Sie wissen, was Sie tun. Am unteren Rand des Einstelldialogs befinden sich drei Schaltflächen: *Zurücksetzen* setzt alle Einstellungen auf

den Standardwert zurück, *Abbruch* macht die aktuellen Änderungen rückgängig und *Anwenden* schließt den Dialog und speichert die Einstellungen.

4.1.1 FITS/SER Debayern

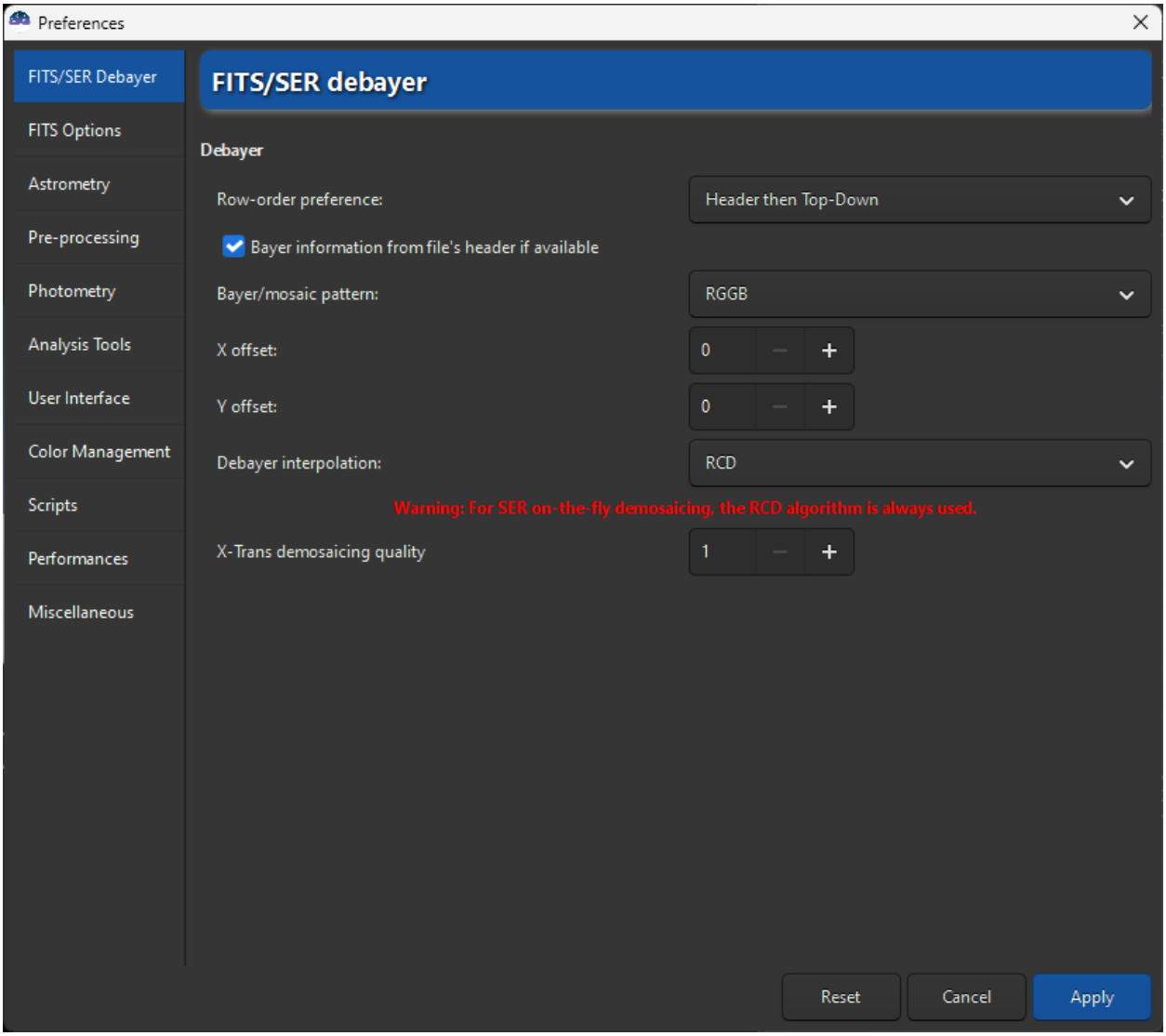


Abb. 1: Seite 1 des Einstellungen-Dialogs

The **FITS/SER debayer** tab allows the user to define the debayer settings for FITS, SER or TIFF files. Consequently, this tab is only usable for a user with an OSC camera. It is advisable to leave the default settings as Siril will automatically define the correct settings to use. However, in the case of a TIFF file that is not an AstroTIFF, or a file that does not have all the required keywords, it may be necessary to adjust the settings manually.

- **Row-order preference**: You can choose to use header *ROW-ORDER* keywword if present and fallback to either Top-Down or Bottom-Up if not present. Or you can force a preferred orientation. More explanation on this keyword can be found *here*.
- **Bayer/mosaic pattern**: This drop-down menu allows you to choose the type of Bayer matrix used by the camera. It is generally indicated in the manufacturer's information. If *Bayer information from file's header if available* is ticked, the value specified in this dropdown will only be used in case the information is not present in the header.

If the option is unticked, it is always used.

- **X offset**: In rare cases, files are recorded with a Bayer array shift. We can define an offset of 1 on the X axis, and an offset of 1 on the Y axis. Here the value defines if there is an offset in X. Note that this is used only when *Bayer information from file's header if available* is unticked or when the header does not contain this information.
- **Y Offset**: Y-Offset der Bayer-Matrix.

Wenn Sie diese Einstellungen ändern, wird jedes Mal ein anderes Demosaicing durchgeführt. Deshalb ist es ratsam, die Einstellungen in der Standardeinstellung zu belassen, es sei denn, Sie sind sich wirklich sicher, was Sie tun.

Eine weitere Option, die sich weniger auf das Endergebnis auswirkt, ist die Wahl des in **Debayer Interpolation** vorgeschlagenen Demosaicing-Algorithmus. Zur Auswahl stehen die folgenden Verfahren:

- **Schnelles Debayern** ist der schnellste in Siril verfügbare Algorithmus. Die anderen unten aufgeführten Algorithmen sind jedoch oft besser.
- **VNG4**, Threshold-Based Variable Number of Gradients (schwellenwertbasierte variable Anzahl von Farbverläufen), arbeitet mit einer 5x5-Pixel-Nachbarschaft um jedes Quellpixel. Dieser Algorithmus eignet sich sehr gut für kontrastarme, flache Bildbereiche (z. B. den Himmelshintergrund), erzeugt aber Artefakte in kontrastreichen Bereichen (z. B. Sterne).
- **AHD**, Adaptive Homogeneity-Directed, ist ein weiterer bekannter Debayer-Algorithmus. Er zeigt jedoch in der Regel Artefakte im Hintergrund und schlechte Sternformen.
- **AMaZE**, Aliasing Minimization and Zipper Elimination, ist ein Algorithmus, der besonders bei rauscharmen Aufnahmen gute Ergebnisse liefert.
- **DCB**, Double Corrected Bilinear, ein neuerer Algorithmus, kann wie AHD einige Artefakte im Hintergrund aufweisen.
- **HPHD**, Heterogeneity-Projection Hard-Decision, ist ein älterer Algorithmus, der gute Ergebnisse liefert, aber recht langsam ist.
- **IGV** und **LMMSE** sind sehr gut für die Arbeit mit stark verrauschten Bildern geeignet. IGV neigt jedoch dazu, einige Farbinformationen zu verlieren, während LMMSE einer der rechenintensivsten Demosaikierer ist und viel Speicherplatz benötigt.
- **RCD**, Ratio Corrected Demosaicing, soll die Farbkorrekturfehler glätten, die bei vielen anderen Interpolationsmethoden üblich sind. Es eignet sich hervorragend für runde Kanten, z. B. bei Sternen, und ist daher der in Siril verwendete Standardalgorithmus.

Für den X-Trans-Sensor wird unabhängig von der in den Einstellungen gewählten Methode ein spezieller Algorithmus namens **Markesteijn** verwendet. Für letzteren ist es möglich, die gewünschte Qualität mit der Option **X-Trans-Demosaikierungsqualität** zu definieren. Sie definiert die Anzahl der Durchläufe für den X-Trans Markesteijn Demosaicing-Algorithmus, 1 ist die Standardeinstellung, 3 kann etwas besser sein, ist aber langsamer.

> **Warnung**
>
> Für das schnelle On-The-Fly-Debayern von **SER**-Dateien wird unabhängig von der Auswahl im Dropdown-Menü immer der RCD-Algorithmus verwendet. Dadurch ist Siril in Bezug auf die Ausführungsgeschwindigkeit effizienter und bietet eine gute Qualität.

4.1.2 FITS Optionen

Die Seite **FITS-Optionen** enthält alle Einstellungen, die sich auf das von Siril verwendete native Dateiformat beziehen.

- **Fits-Dateierweiterung**: Standardmäßig ist der Wert auf **.fit** eingestellt. Viele Aufnahmeprogramme verwenden jedoch die Erweiterung fits. In diesem Fall raten wir Ihnen, den Wert anzupassen. Alle von Siril erstellten Dateien werden die hier definierte Erweiterung haben. Außerdem können nur Sequenzen mit der in den Einstellungen

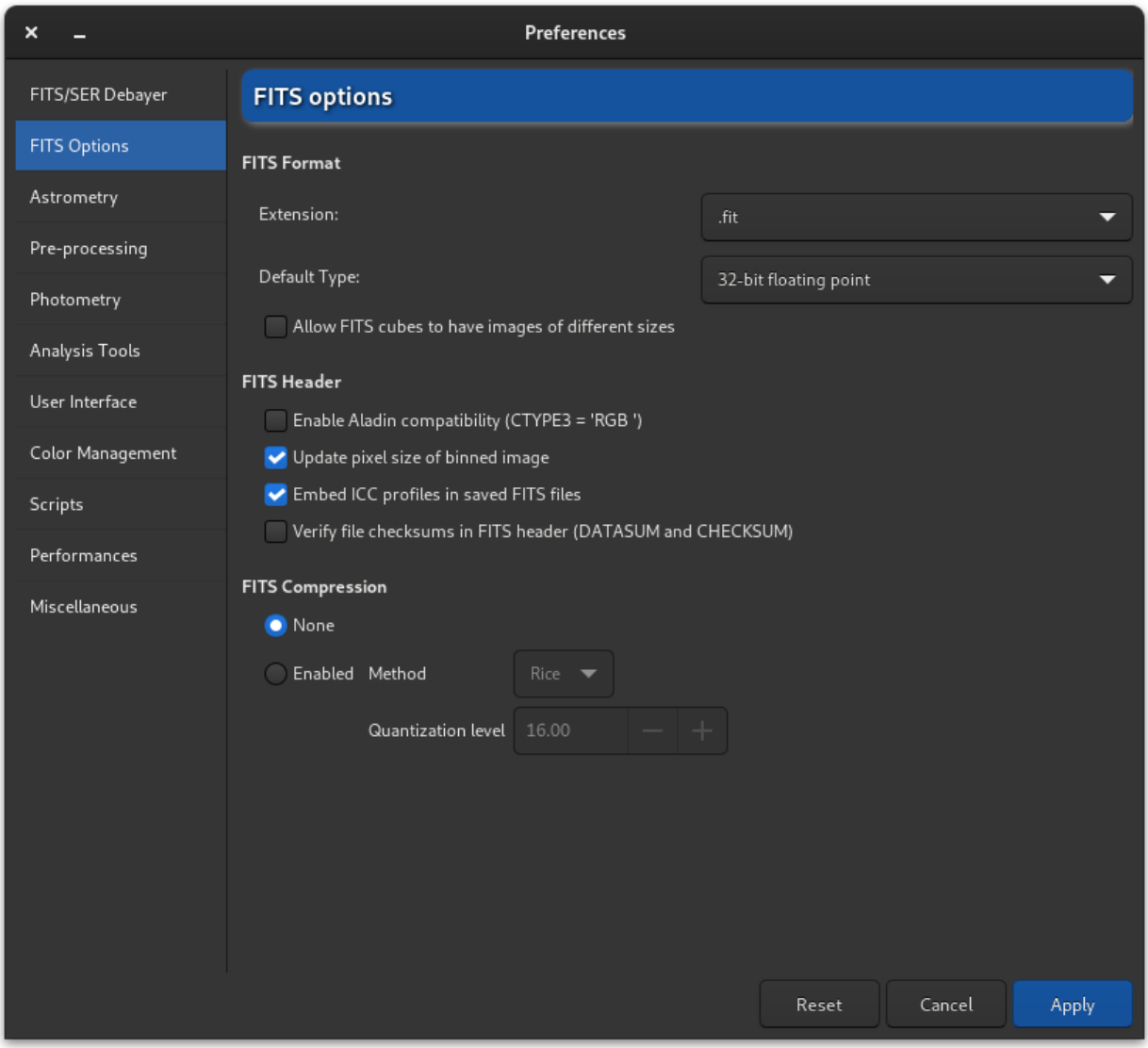


Abb. 2: Seite 2 des Einstellungs-Dialogs

festgelegten Erweiterung geladen werden. Es ist daher nicht möglich, eine **.fits**-Sequenz und eine **.fit**-Sequenz zu öffnen, ohne diesen Wert zu aktualisieren. Unterstützte Erweiterungen sind:

- **.fit**
- **.fits**
- **.fts**

Sie können alle mit der Erweiterung **.fz** versehen werden, wenn die Daten komprimiert sind.

Siril Kommandozeile

```
setext extension
```

Legt die verwendete und von Sequenzen erkannte Erweiterung fest.

Das Argument **Erweiterung** kann "fit", "fts" oder "fits" sein

- **Standardeinstellung**: Standardmäßig arbeitet Siril mit 32-Bit-Fließkommazahlen im Bereich [0, 1]. Dies ist der beste Weg, um eine hohe Genauigkeit zu erreichen. Aus Gründen des Speicherplatzes auf der Festplatte kann ein Benutzer jedoch auch mit 16-Bit-Ganzzahlen ohne Vorzeichen (im Bereich [0, 65535]) arbeiten. Seien Sie jedoch vorsichtig, denn bei einem 16-Bit-Stacking können viele Informationen verloren gehen.
- **Erlaubt FITS Würfeln unterschiedliche Größe**: Dies kann nützlich sein, um wissenschaftliche FITS-Dateien zu öffnen, die nicht von Siril erstellt wurden und die mehrere Bilder unterschiedlicher Größe enthalten, die sonst als ungültige Siril FITSEQ-Dateien betrachtet würden. Die JWST-Bilder sind ein gutes Beispiel für die Verwendung dieser Option. Ein Beispiel finden Sie *hier*.
- **Kompatibilität für Aladin einschalten (CTYPE3 = 'RGB')**: Aladin betrachtet einen 3D-FITS-Würfel als RGB-Bild (Rot-, Blau- und Grün-Komponenten), wenn das FITS-Schlüsselwort `CTYPE3 = 'RGB '` im Header angegeben ist. In diesem Fall werden alle `BITPIX` Werte unterstützt. Ohne die Angabe des FITS-Schlüsselwortes `CTYPE3 = 'RGB '` werden nur FITS-Würfel mit 3 Frames in der gleichen Dimension und mit einem `BITPIX=8` automatisch als RGB-FITS erkannt.

Warnung

Diese Option kann zu Konflikten mit der Astrometrie-Funktion führen und sollte nur aktiviert werden, wenn sie wirklich notwendig ist.

- **Aktualisierung der Pixelgröße von gebinnten Bildern**: Für die Berechnung der Abtastrate kann die Pixelgröße auf zwei verschiedene Arten angegeben werden: die tatsächliche Pixelgröße wird angegeben, muss aber mit dem Binning multipliziert werden (wenn angekreuzt), die bereits multiplizierte Pixelgröße wird angegeben (wenn nicht angekreuzt). Dies hängt von der Erfassungssoftware ab, die zur Erstellung der FITS-Datei verwendet wird.
- **ICC-Profile in gespeicherte FITS-Dateien einbetten**: Diese Option sollte grundsätzlich aktiviert bleiben, es sei denn, Sie benötigen die Interoperabilität mit einem Programm, das FITS mit eingebetteten ICC-Profilen nicht verarbeiten kann (derzeit sind keine bekannt).
- **Dateiprüfsummen im FITS-Header (DATASUM und CHECKSUM) überprüfen**: Überprüfen Sie Dateiprüfsummen, falls vorhanden. Die Überprüfung erfolgt bei jedem Lesen der Datei, was die Leistung von Siril bei Verwendung einer Sequenz erheblich beeinträchtigen kann.
- **FITS Kompression**: Die Komprimierung kann in einigen Fällen interessant sein, in denen der Speicherplatz auf der Festplatte ein entscheidender Punkt bei der Verarbeitung ist. Weitere Informationen finden Sie in dem

Abschnitt, der dem FITS-Format gewidmet ist, *hier*.

Die Komprimierung fügt den erzeugten Dateien die Erweiterung `.fz` hinzu. Siril ist in der Lage, eine Sequenz mit der Erweiterung `.fz` zu öffnen, ohne einen Wert in den Voreinstellungen ändern zu müssen.

Siril Kommandozeile

```
setcompress 0/1 [-type=] [q]
```

Legt fest, ob Bilder komprimiert werden oder nicht.

0 bedeutet keine Kompression, während **1** die Kompression aktiviert.
Wenn die Komprimierung aktiviert ist, muss der Typ ausdrücklich in der Option **-type=** angegeben werden ("rice", "gzip1", "gzip2").
Im Zusammenhang mit der Komprimierung muss der Quantisierungswert im Bereich [0, 256] liegen.

Beispiel: "setcompress 1 -type=rice 16" stellt die Rice-Kompression mit einer Quantisierung von 16 ein

4.1.3 Astrometrie

Tipp

Vergessen Sie nicht, dass Sie im Fenster scrollen können, da es unten Optionen gibt.

Diese Registerkarte enthält alle Optionen im Zusammenhang mit der Astrometrie. Die Astrometrie ist eine in Siril stark implementierte Funktion. Wenn das Bild astrometrisch gelöst ist (d.h. wenn die Astrometrie erfolgreich war), ist es möglich, die Namen der bekannten Objekte anzuzeigen. Insbesondere diejenigen, die in den großen astronomischen Katalogen aufgeführt sind. Im Teil "Beschriftungen" kann man festlegen, welche Kataloge für die Anzeige der Beschriftungen verwendet werden können. Derzeit gibt es 8 davon, und sie können abgewählt werden, um ignoriert zu werden:

- Messier katalog
- New General Catalogue (NGC)
- Index Catalogue (IC)
- Lynds' Catalogue of Bright Nebulae (LBN)
- Sharpless Catalogue (Sh2)
- Catalogue of Brightest stars
- IAU Sternbildlinien
- IAU constellations names (positions collected from this page)

Zusätzlich zu dieser Liste gibt es zwei weitere Kataloge, die vom Benutzer ausgefüllt werden können. Einer betrifft die Deep-Sky-Objekte, der andere das Sonnensystem. Sie können im Abschnitt *Beschriftungen* dieser Dokumentation besser beschrieben werden.

Unter der Liste der astronomischen Kataloge gibt es eine Option zum Festlegen eines bevorzugten IAU-Observatoriumscodes. Dies ist ein 3-stelliger Code, mit dem ein nahegelegenes IAU-Observatorium angegeben werden kann, um die Standortgenauigkeit beim Ausführen von „conesearch -cat=solsys“ zu verbessern. Wenn die Option

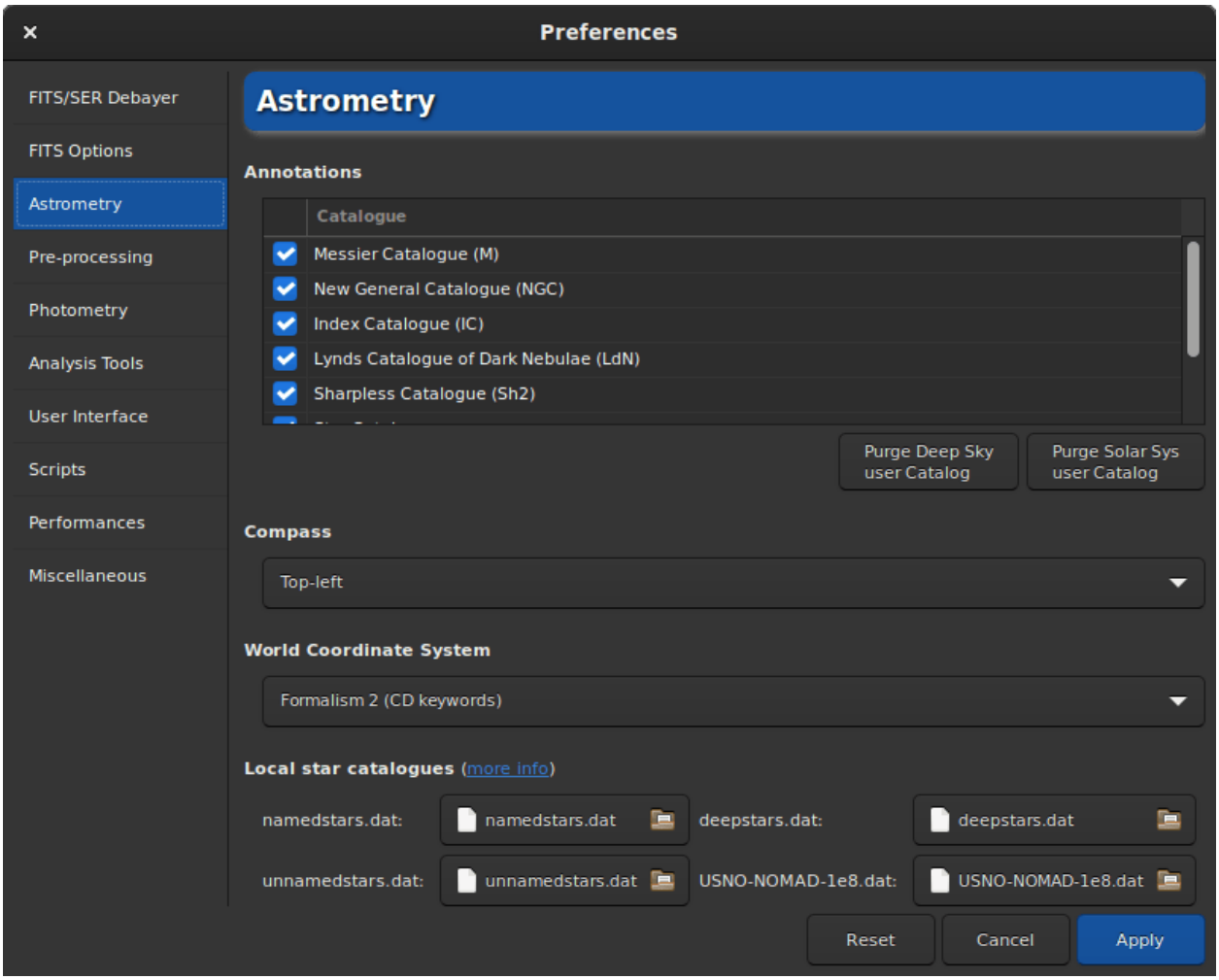


Abb. 3: Seite 3 des Einstellungs-Dialogs

„-obscode=“ nicht angegeben ist, wird dieser Standardwert anstelle des andernfalls verwendeten geozentrischen Standardwerts verwendet. Der Optionstext enthält einen Hyperlink zur Website des Minor Planets Center, auf der eine aktuelle Liste der IAU-Observatoriumscodes verfügbar ist, sodass Sie den Code, der Ihrem Standort am nächsten liegt, leicht finden können.

Durch Klicken auf die Schaltfläche *Objektnamen anzeigen* (nur, wenn das Bild eine astrometrische Lösung hat), werden die Beschriftungen auf dem Bild angezeigt. Es ist auch möglich, auf die Schaltfläche zur Anzeige des Himmels-Koordinatensystems zu klicken. Letzteres fügt standardmäßig einen Kompass in die Mitte des Bildes ein. Im Abschnitt **Kompass** können Sie die gewünschte Position für die Anzeige des Kompasses festlegen.

Der Abschnitt **Weltsystemkoordinaten** erlaubt es folgendes zu wählen

- **Formalismus 1 (CD-Stichwörter)**: Im `PC i_j`-Formalismus werden die Matrixelemente m_{ij} (lineare Transformationsmatrix) in `PC i_j` (fließend-wertig) Header-Karten und si als `CDELT i` kodiert. Die *i* und *j* Indizes werden ohne führende Nullen verwendet, z.B. `PC 1_1` und `CDELT 1`. Die Standardwerte für `PC i_j` sind 1.0 für $i = j$ und 0.0 ansonsten. Die `PC i_j`-Matrix darf nicht singulär sein; sie muss eine Inverse haben. Außerdem müssen alle `CDELT i` ungleich Null sein.
- **Formalismus 2 (PC-Stichwörter)**: Die Schlüsselwörter `CD i_j` (mit Fließkommazahlen) kodieren das Produkt $s_i m_{ij}$. Die *i* und *j* Indizes werden ohne führende Nullen verwendet, z.B. `CD 1_1`. Die Matrix `CD i_j` darf nicht singulär sein; sie muss eine Inverse haben. `CDELT i` und `CROTA i` dürfen mit `CD i_j` koexistieren, um alten FITS-Interpretern zu helfen, müssen aber von neuen Lesern ignoriert werden.

Der Teil **Lokale Sternkataloge** des Dialogfensters betrifft die Verwendung lokaler Kataloge zur astrometrischen Lösung von Bildern. Diese Funktion wird ausführlich im Abschnitt *Astrometrische Lösung* dieser Dokumentation beschrieben.

Im Abschnitt **Allgemeine astrometrische Lösung** stehen eine Reihe von Optionen zur Steuerung sowohl des internen Siril-Lösers als auch von astrometry.net zur Verfügung.

- Die erste Option legt fest, ob die berechnete Brennweite und die Eingabepixelgröße in den Einstellungen als Standardwerte für Bilder gespeichert werden, die nicht über die entsprechenden Metadaten verfügen, wenn eine astrometrische Lösung gefunden wird.
- **Abtastratentoleranz**: Prozentsatz, der unterhalb und oberhalb der erwarteten Abtastrate liegen darf. Die gegebene Abtastrate wird mit 1 + dies / 100 multipliziert oder geteilt.
- **Grade der Polynomkorrektur**: Astrometrische Löser können eine Polynomkorrektur (SIP) verwenden, um mit optischen Aberrationen/Verzerrungen zu arbeiten. Dies ist die Ordnung des Polynommodells. 0 deaktiviert es.
- **Zielradius**: erlaubter Suchradius um die Zielkoordinaten für die Lösung (Grad). Unbenutzt für blinde (kein Ziel übergeben) Lösungen.
- **Maximale Sekunden für den Lösungsversuch**: erlaubte Zeit für die Lösung für jede Katalogdatei. Sie kann nur dann als Gesamtlösungszeit verwendet werden, wenn solve-field in seiner Konfigurationsdatei so konfiguriert ist.

Der letzte Abschnitt ist dem solve-field Plate-Solver aus der astrometry.net Suite gewidmet.

- **,xyls-Dateien nicht löschen**: Die Liste der Sterne wird als FITS-Tabelle an solve-field übergeben, aktivieren Sie dies, damit die Datei im Arbeitsverzeichnis bleibt.
- **.wcs-Dateien nicht löschen**: Die Ergebnisse von solve-field werden in einem FITS-Header mit einem Dateinamen gespeichert, der auf .wcs endet. Aktivieren Sie diese Option, um diese Datei nicht zu entfernen.
- **Verzeichnis des lokalen astrometry.net-Lösers**: Um Astrometry.net lokal in Siril nutzen zu können, kann es erforderlich sein, Siril den Pfad mitzuteilen, auf dem er installiert wurde. Auf UNIX-basierten Systemen befindet es sich im Allgemeinen in der Variablen „PATH“ und ist nicht erforderlich. Wenn Sie unter Windows das

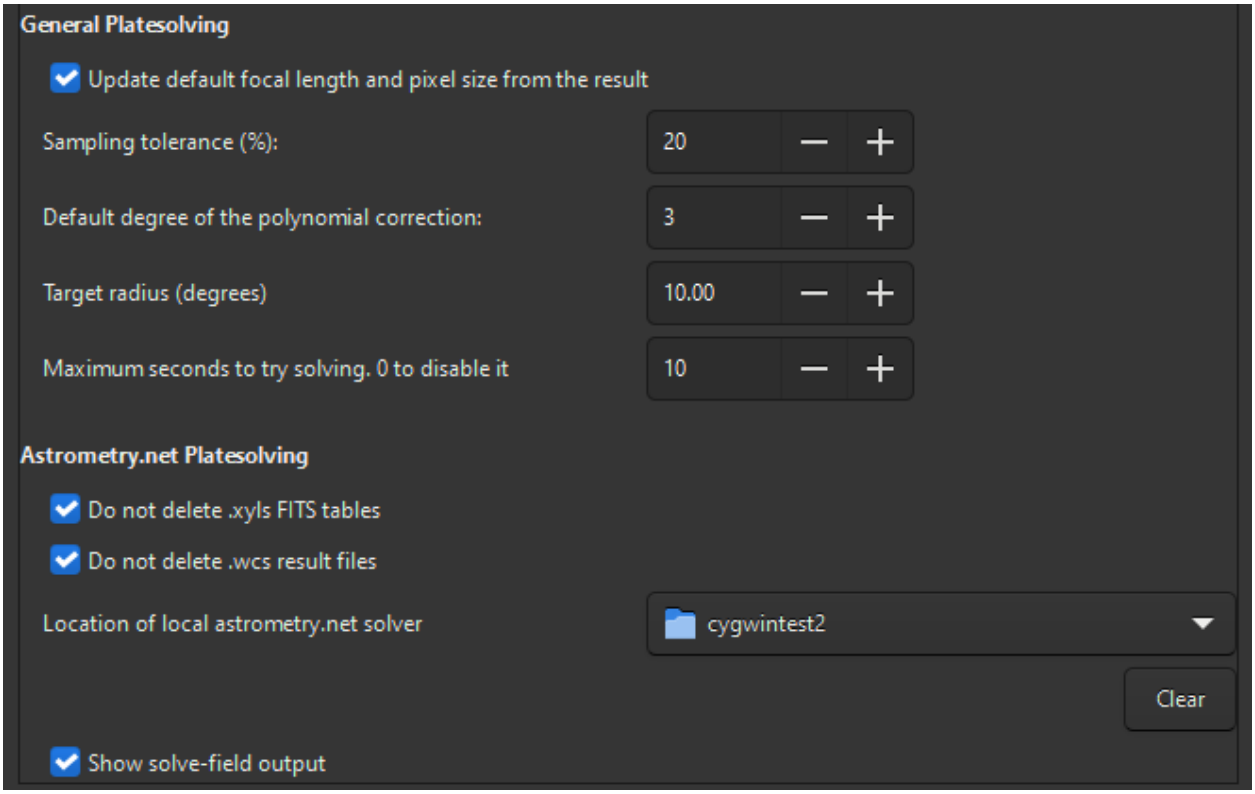


Abb. 4: Unterer Teil auf Seite 3 des Einstellungsdialogs

Standardinstallationsverzeichnis, also „%LOCALAPPDATA%\cygwin_ansvr“, nicht geändert haben, sucht Siril ohne zusätzliche Konfiguration danach. Wenn Sie Cygwin haben und astrometry.net aus den Quellen erstellt haben, müssen Sie hier das Cygwin-Stammverzeichnis angeben.

- **Ausgabe von solve-field anzeigen**: gibt die Ausgabe des Solvers im Hauptprotokollfenster von Siril aus, andernfalls wird nur das Ergebnis angezeigt.

4.1.4 Vorverarbeitung

Die Registerkarte Pre-Processing enthält alle Elemente im Zusammenhang mit den Schritten, die bis vor dem Stacking ausgeführt werden. Hier können Sie eine Bibliothek mit Offsets, Dunkelbildern, Flachbildern und Verzerrungen, den Ausgabennamen der gestapelten Datei oder spezifische Korrekturen für Kameras, die den X-Trans-Sensor verwenden, verwalten.

- **Dark/Bias/Flat/distortion Libraries**: In this section it is possible to load an offset, a dark and a flat that will be used by default in the pre-processing if the button to the right of the text box, *Use it as default* is checked. Each path will also be stored in the reserved keywords `$defbias`, `$defdark`, `$defflat` and `$defdisto` (one token `$`) which can be used when saving a stacking/platesolving result. As far as bias is concerned, it is possible to use more than just a file path. Indeed, in the Siril team we encourage users to use synthetic bias as explained in this tutorial. Several values are then possible as long as the first character entered is the = sign. It is possible to use a fixed integer value like `=500` or a multiplication involving the keyword `$OFFSET` (one token `$`) as long as the latter is actually registered in the FITS file header, like `10*$OFFSET`. More details are given in the tutorial.
- **Stacking Standardeinstellungen**: Hier legen wir den Standardnamen fest, den wir den Stacking-Ergebnissen geben wollen. Es ist möglich, jeden Wert, der im FITS-Header angegeben ist, als Schlüsselwort zu verwenden und ihn mit `$`-Token zu umgeben. Wenn das Schlüsselwort nicht existiert, wird die Variable verwendet, andernfalls ist es ihr Wert. Ein weiteres reserviertes Schlüsselwort, das verwendet werden kann,

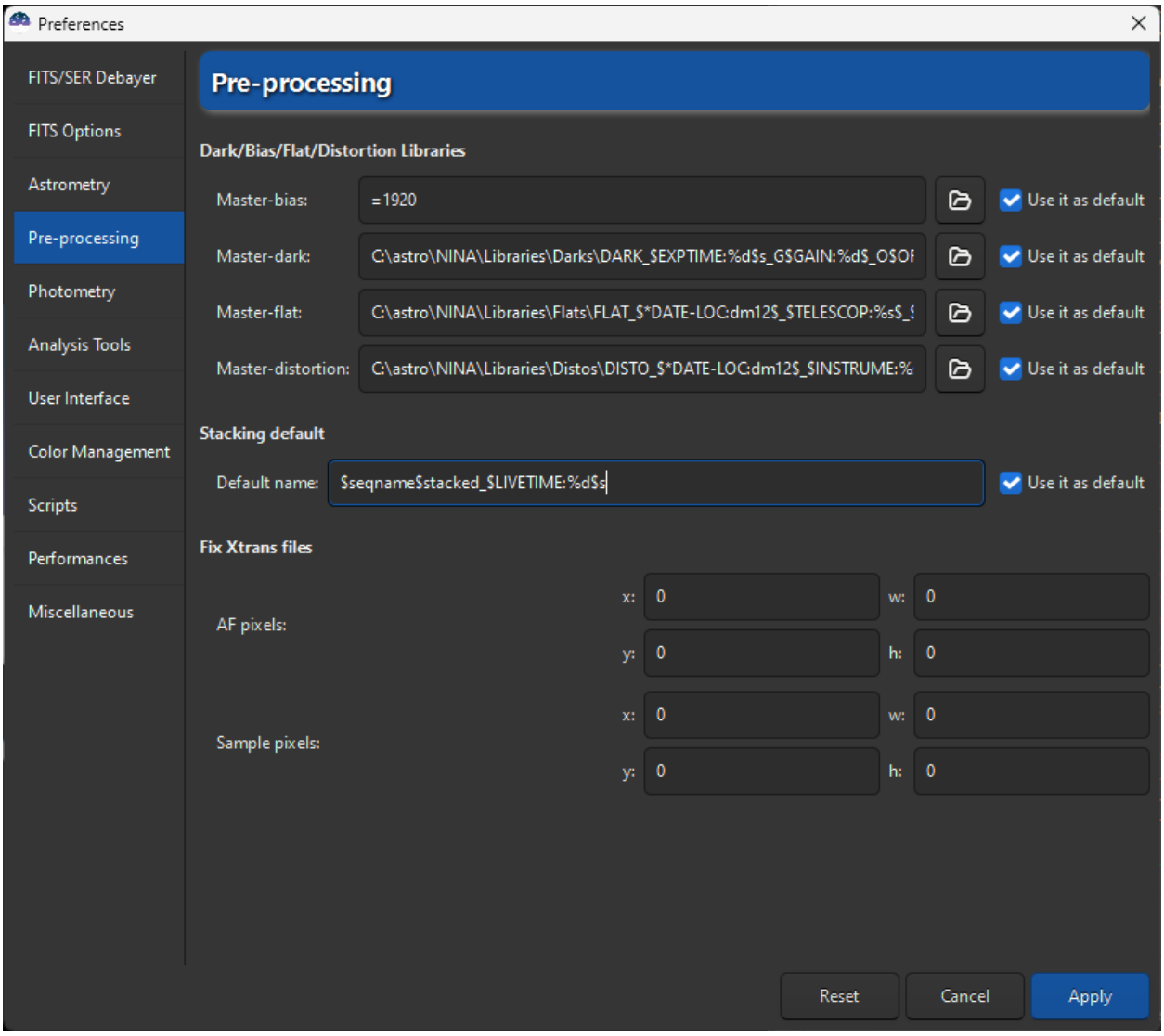


Abb. 5: Seite 4 des Einstellungs-Dialogs

ist `$seqname$`. Es enthält den Namen der geladenen Sequenz. Zum Beispiel, der folgende Standardname, `$seqname$stacked_$LIVETIME:%d$s` mit einem Sequenznamen `r_pp_light_` und dem folgenden FITS-Header:

```
...
DATE    = '2022-12-08T22:21:14' / UTC date that FITS file was created
DATE-OBS= '2015-08-21T22:18:25' / YYYY-MM-DDThh:mm:ss observation start, UT
STACKCNT=                   13 / Stack frames
EXPTIME =                 300. / Exposure time [s]
LIVETIME=                3900. / Exposure time after deadtime correction
EXPSTART=     2457256.42945602 / Exposure start time (standard Julian date)
EXPEND  =     2457256.51666667 / Exposure end time (standard Julian date)
...
```

wird `r_pp_light_stacked_3900s.fit` ausgegeben.

Bemerkung

Sowohl Bibliothekspfade als auch Stacking unterstützen standardmäßig *Pfadauswertung*.

- **Xtrans-Dateien reparieren**: Dieses Einstellungsfeld ist sehr spezifisch und betrifft nur die Besitzer bestimmter X-Trans-Sensoren. In der Tat zeigen einige Bilder von diesen Kameras ein großes Quadrat in der Mitte der dunklen und verzerrten Bilder aufgrund der Position des Autofokus (AF). Siril verfügt über einen Algorithmus zur Beseitigung dieses Problems für die folgenden Kameras:
  - Fujifilm X-T1
  - Fujifilm X-T2
  - Fujifilm X-T20
  - Fujifilm X-Pro2
  - Fujifilm X-E3
  - Fujifilm X-H1

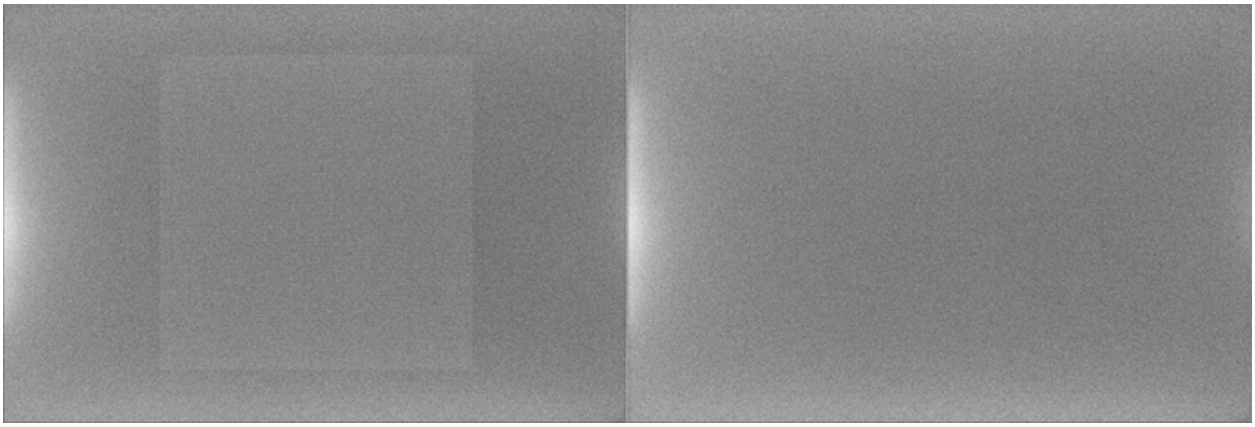


Abb. 6: Behebung des X-Trans-Artefakts durch den Algorithmus von Siril

Für den unwahrscheinlichen Fall, dass Ihre Kamera dieses Artefakt enthält und nicht unterstützt wird, ist es möglich, die anzuwendende Korrektur hier zu definieren. Am besten wenden Sie sich an das Entwicklerteam, um die Werte für Ihre Kamera einzugeben.

4.1.5 Photometrie

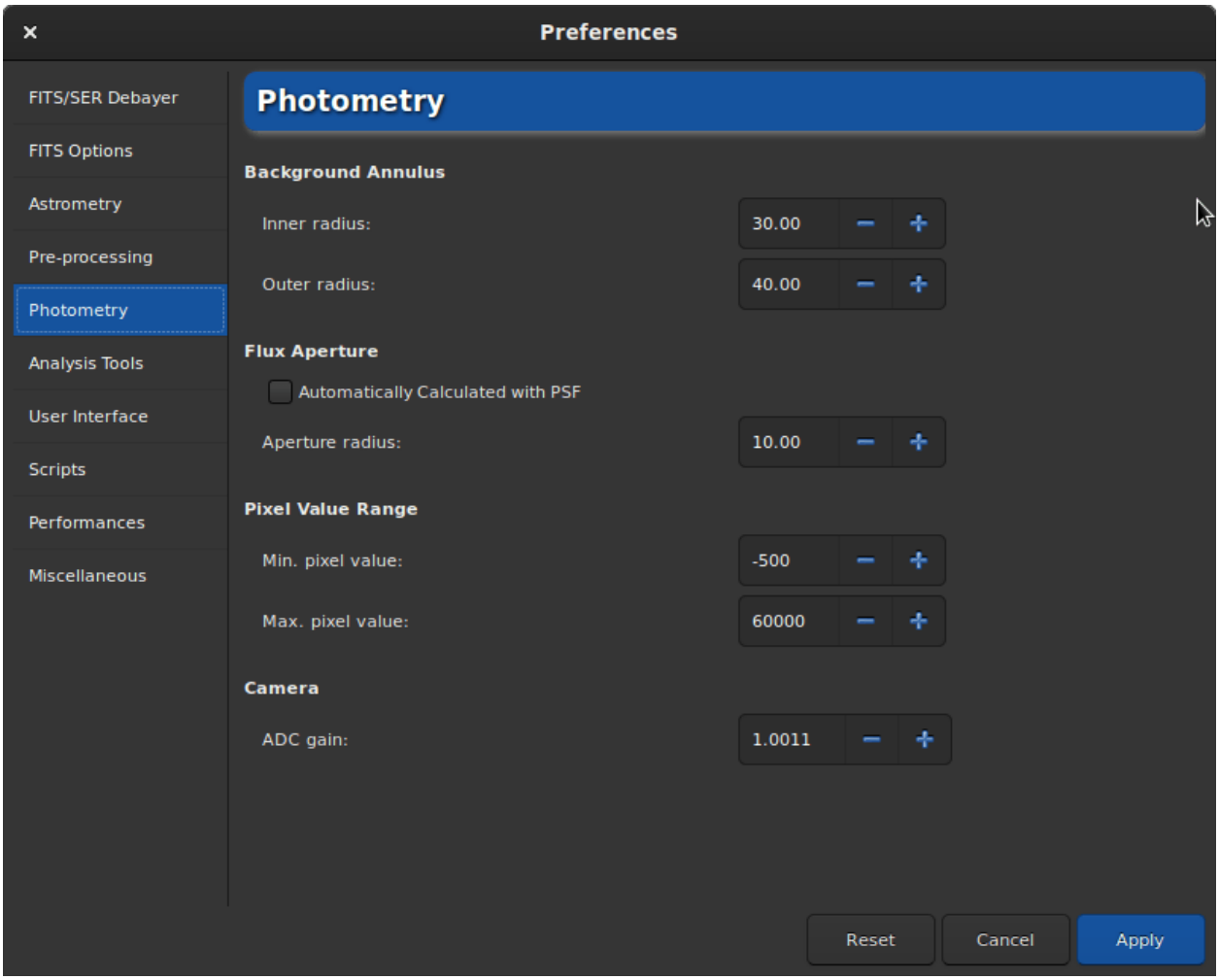


Abb. 7: Seite 5 des Einstellungs-Dialogs

Die Photometrie, d.h. die Untersuchung des Lichts, ist eine weitere Funktion, die in Siril sehr präsent ist. In diesem Abschnitt der Voreinstellungen können Sie die mit diesem Werkzeug verbundenen Einstellungen festlegen.

Das Grundprinzip der Aperturphotometrie besteht darin, den beobachteten Lichtstrom in einem bestimmten Radius um das Zentrum eines Objekts zu summieren und dann den Gesamtbeitrag des Himmelshintergrunds in derselben Region (berechnet im Ring zwischen dem inneren und dem äußeren Radius) abzuziehen, so dass nur der Lichtstrom des Objekts übrig bleibt, um eine instrumentelle Helligkeit zu berechnen. Dies wird im Abschnitt *Photometrie* dieser Dokumentation genauer beschrieben.

- Es ist dann möglich, den **Inneren Radius** und den **Äußeren Radius** zu ändern, um eine Größe zu definieren, die den berechneten Himmelswert optimiert, wobei versucht wird, die Sterne im Inneren des Rings zu vermeiden. Der äußere Radius **muss** immer **größer** sein als der innere Radius. Standardmäßig ist der Radius der **Flussblende** auf das Doppelte der FWHM der PSF eingestellt, es ist jedoch möglich, diesen Automatismus zu deaktivieren und einen festen Wert manuell zu definieren.
- Mit dem **Pixel-Wertebereich** kann der Benutzer einen Grenzwert festlegen, ab dem ein Pixel als schlecht für die Photometrie angesehen wird. In der Tat wird die Photometrie bei gesättigten Daten niemals gute Ergebnisse liefern, aber auch die Annäherung an hohe Werte kann ungeeignet sein, da sie in den nichtlinearen Bereich der Sensoren fallen kann. Um diesen Bereich zu vermeiden, wurde ein Standardwert von 50000 ADU festgelegt, der

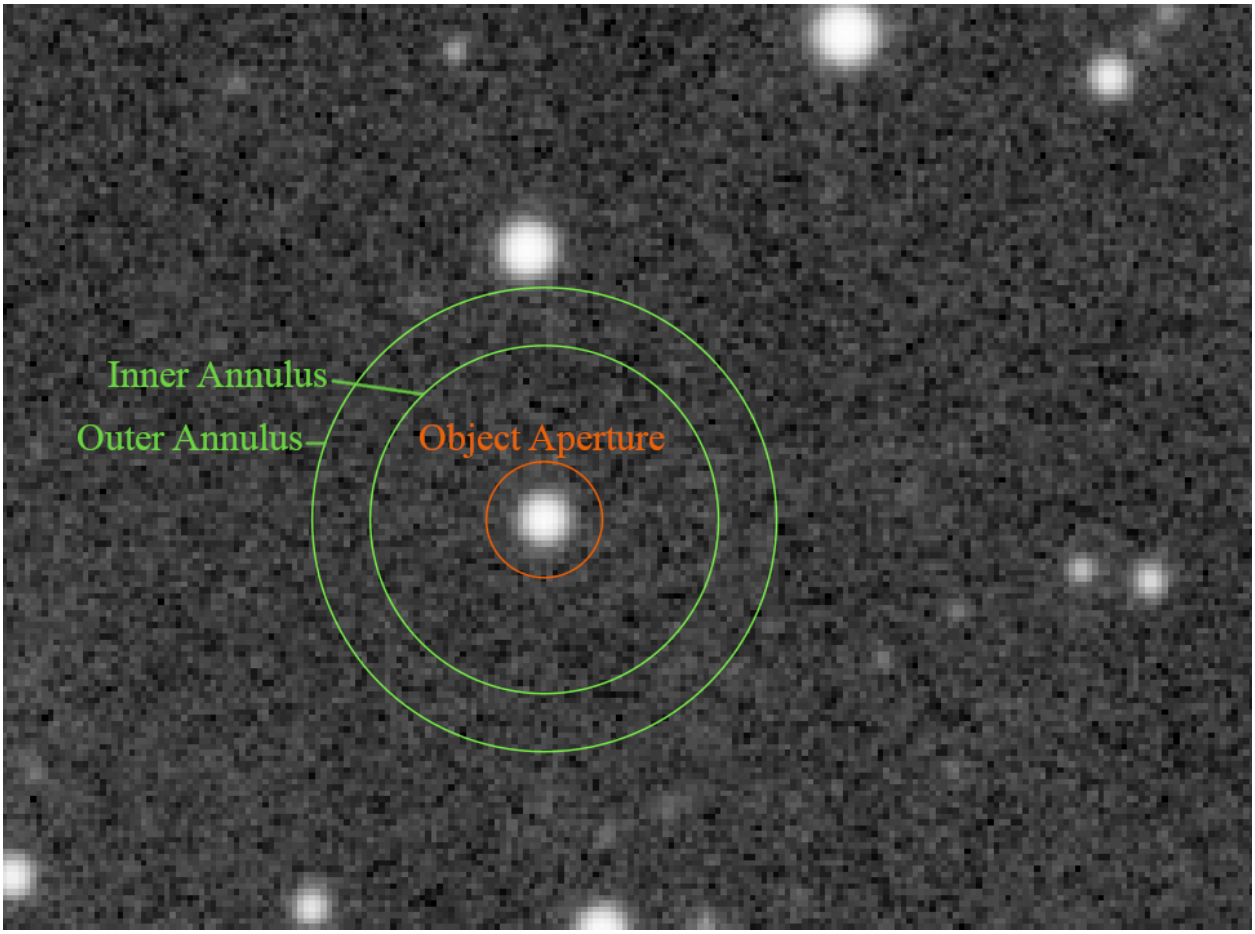


Abb. 8: Kreis der Aperturphotometrie/Hintergrundwert

jedoch von Sensor zu Sensor variieren kann. Negative Werte sind ebenfalls zulässig, da sich das Rauschen im Durchschnitt um einen positiven Wert herum bewegen kann, aber immer noch einige Pixel mit negativen Werten vorhanden sind.

- Schließlich wird dringend empfohlen, falls bekannt, den Wert der A/D-Wandler-Verstärkung** in Elektronen pro ADU anzugeben: Er wird bei den Berechnung der Unschärfe verwendet, falls er nicht bereits im Header der verarbeiteten Bilder angegeben ist.

4.1.6 Analysewerkzeuge

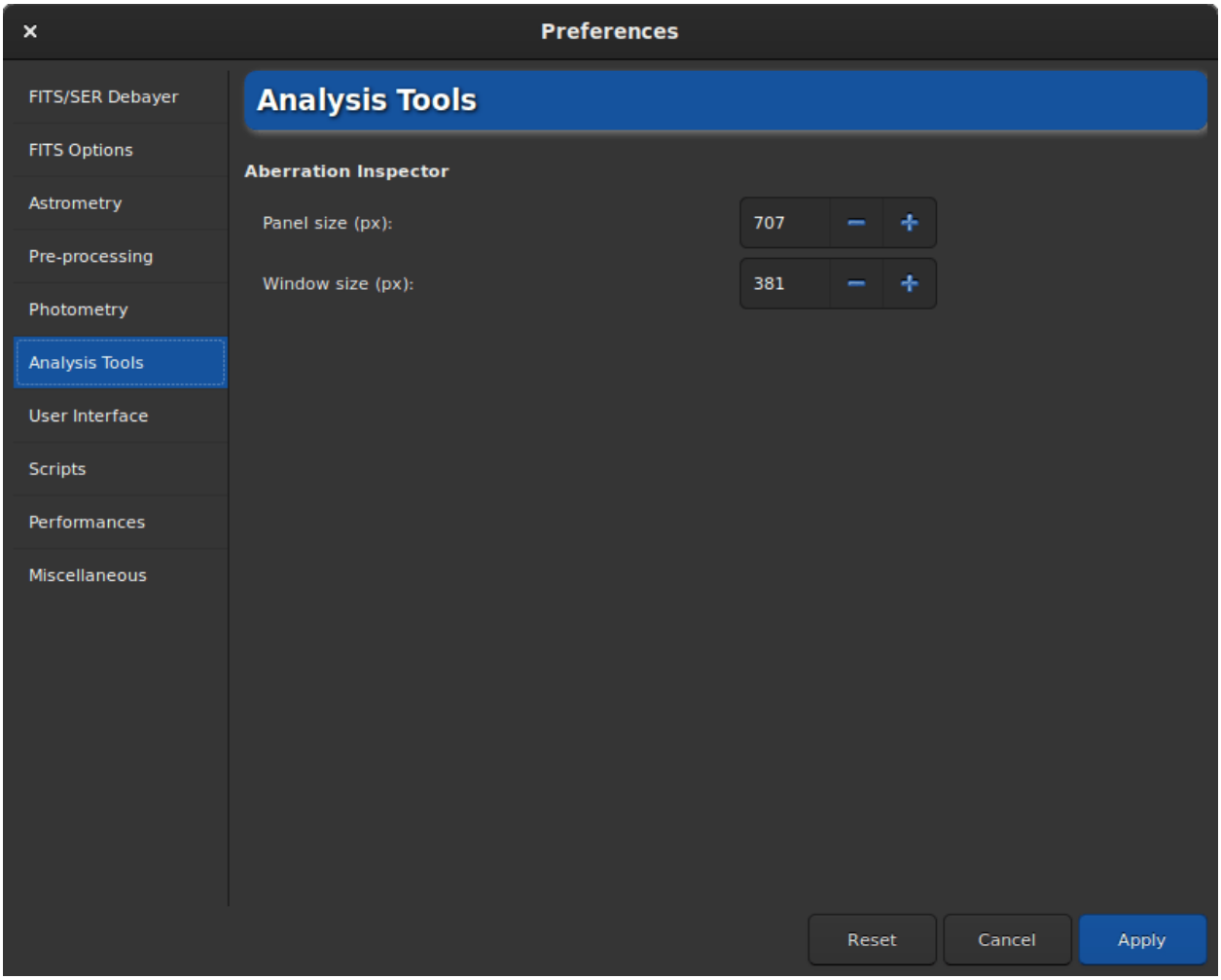


Abb. 9: Seite 6 des Einstellungs-Dialogs

Bisher erfordert nur ein Bildanalysewerkzeug Einstellparameter. Es ist das Werkzeug Aberrationsinspektor. Auf dieser Registerkarte können Sie Einstellungen vornehmen:

- Die **Panelgröße**, in Pixeln, die die Größe des Bildes definiert, das in einem Panel platziert wird. Je größer der Wert, desto größer ist das Bild in einem Panel. Ein zu hoher Wert kann dazu führen, dass die Defekte der Sterne nicht mehr sichtbar sind.
- Die **Fenstergröße**, ebenfalls in Pixel, die die Größe des Dialogs definiert. Es ist normalerweise eine gute Idee, diesen Wert zu erhöhen, wenn ein 4K-Bildschirm verwendet wird.

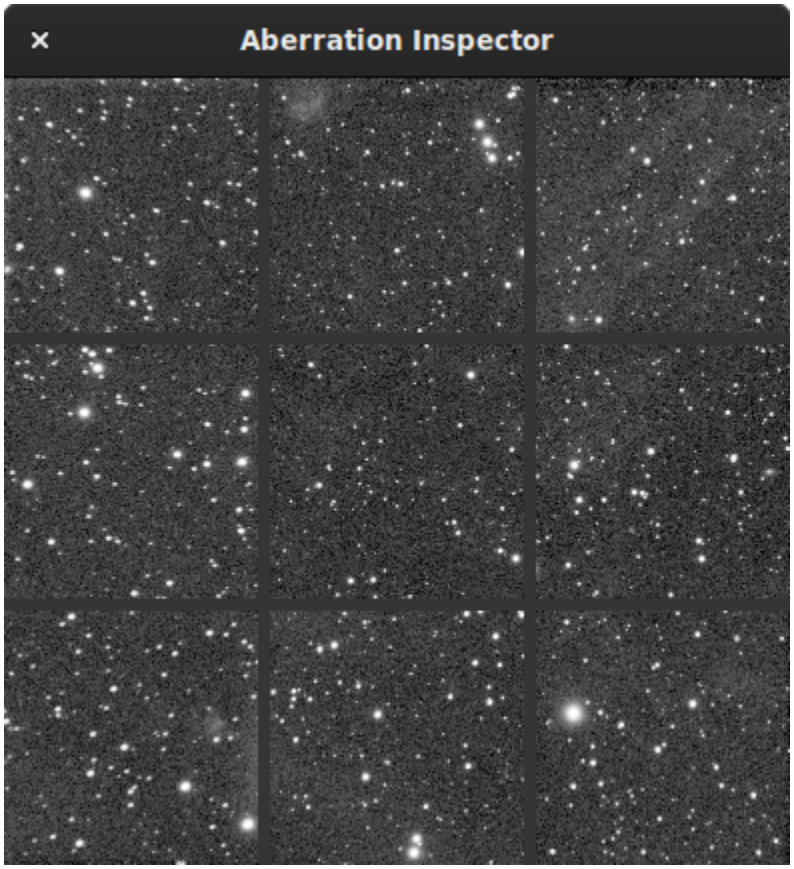


Abb. 10: Fenster Aberration Inspektor

4.1.7 Benutzeroberfläche

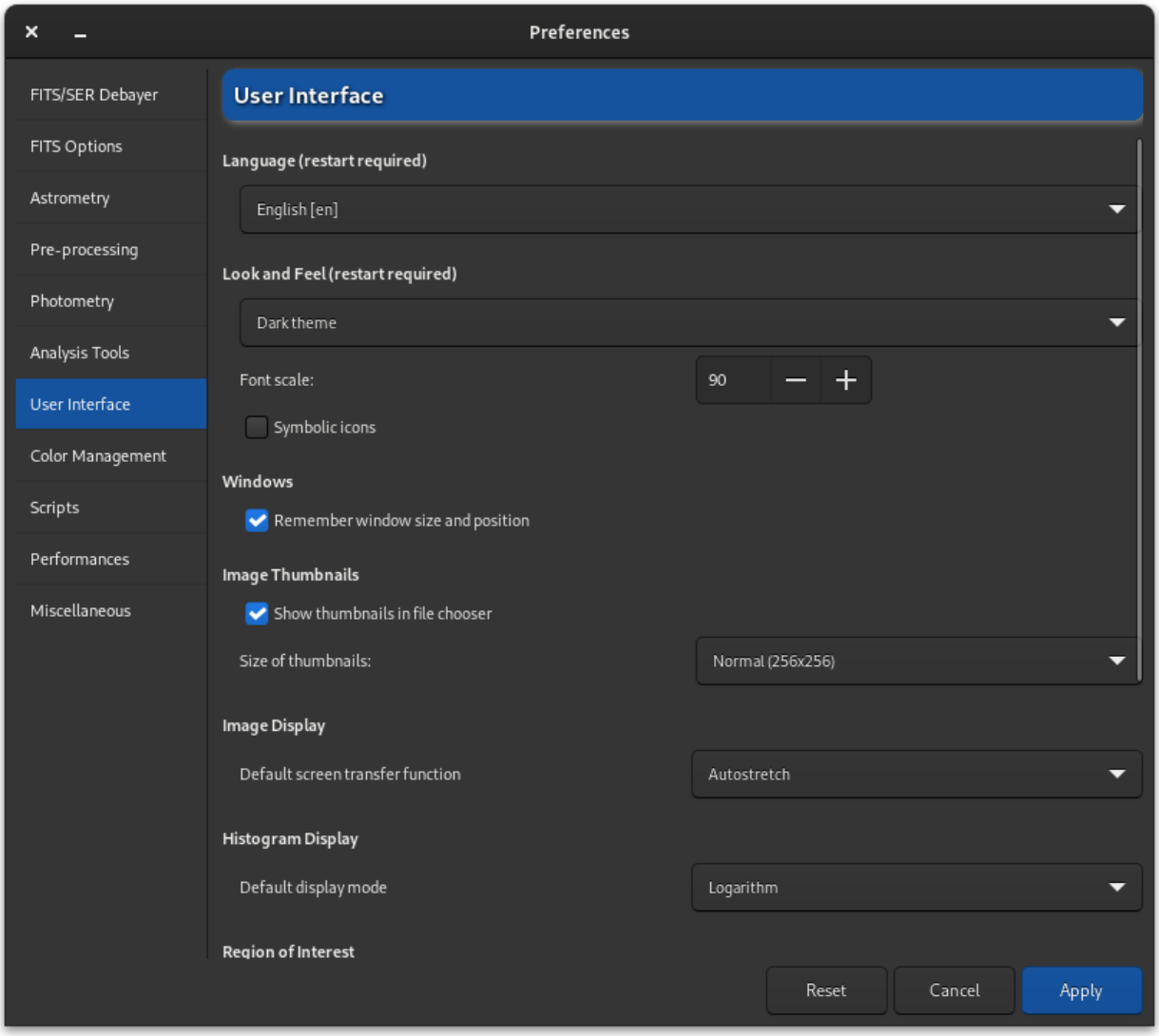


Abb. 11: Seite 7 des Einstellungs-Dialogs

Auf dieser Registerkarte sind alle Einstellungen aufgeführt, die die Benutzeroberfläche betreffen. Dabei handelt es sich nicht um Einstellungen, die sich auf die Prozesse auswirken, sondern auf das "Look and Feel" und die Bedürfnisse des Benutzers.

> **Tipp**
>
> Vergessen Sie nicht, dass Sie im Fenster scrollen können, da es unten Optionen gibt.

- Standardmäßig ist die **Sprache** von Siril entsprechend der Systemsprache eingestellt. Es ist jedoch möglich, die Sprache zu ändern und sie nach Ihren Bedürfnissen einzustellen, solange sie existiert. Beachten Sie jedoch, dass Siril in englischer Sprache entwickelt wird.
- Es sind zwei Themen verfügbar:
  - Das dunkle Thema (Standard)

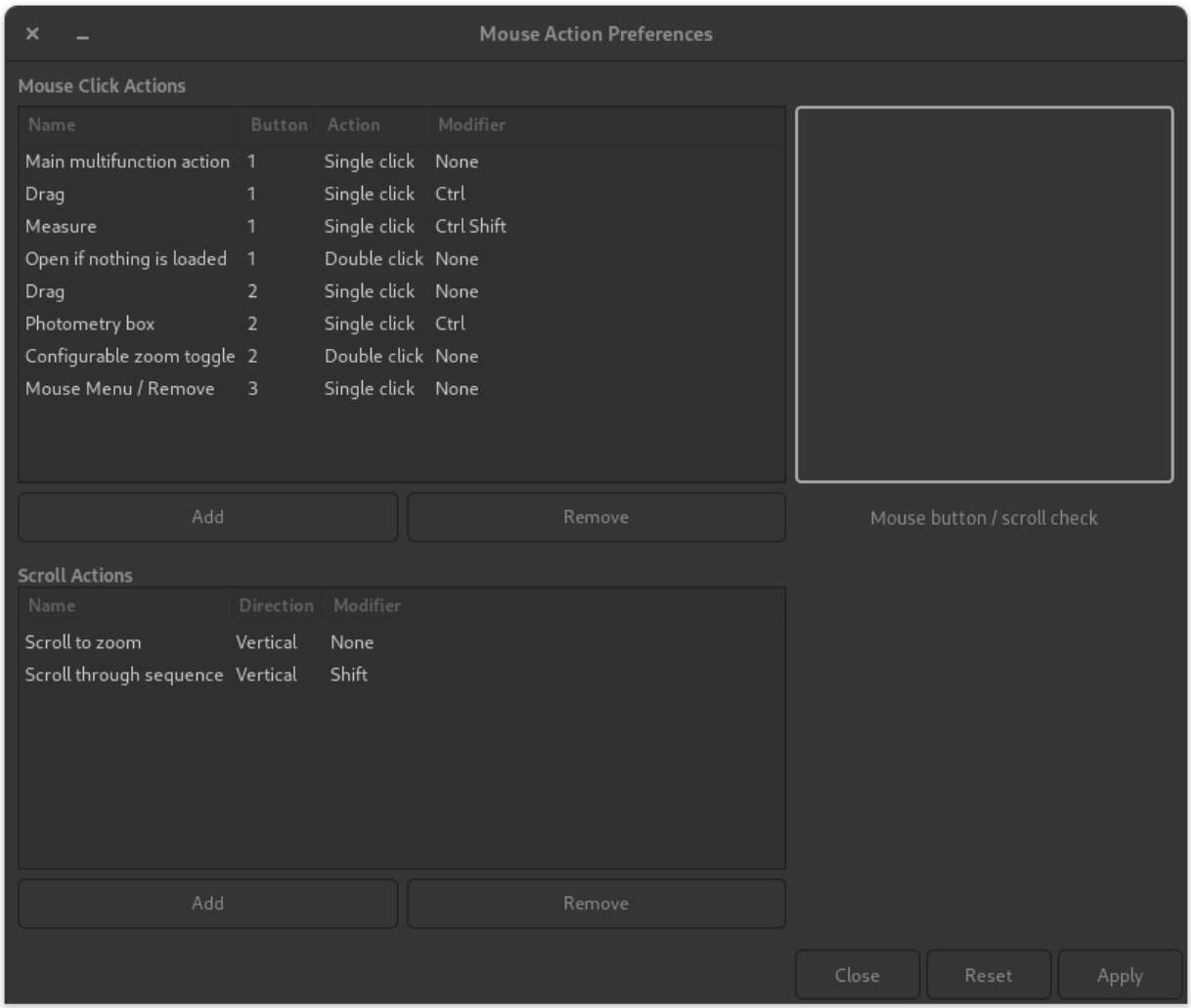
  - Das helle Thema

  Der Wechsel des Themas erfordert einen Neustart der Anwendung, damit sie voll funktionsfähig ist.

- Es ist möglich, die **Schriftgröße** für Benutzer mit einem 4K-Ultra-HD-Bildschirm anzupassen oder **symbolische Symbole** für einige Symbole zu verwenden. Diese Einstellungen erfordern auch einen Neustart der Anwendung.
- Standardmäßig merkt sich Siril die Größe und Position des Anwendungsfensters, wenn Sie es schließen. Durch Aktivieren der Schaltfläche *Fenstergröße und -position speichern* können Sie dieses Verhalten deaktivieren.
- Die Vorschaubilder der Dateien sind normalerweise in den geöffneten Dialogfeldern sichtbar. Die Voreinstellungen erlauben es, sie nicht anzuzeigen, wenn der Computer nur über eine begrenzte Leistung verfügt und der Benutzer keine Notwendigkeit dafür sieht. Über die Dropdown-Liste können Sie auch die Größe der Vorschaubilder ändern.
- **Standard-Bildschirmübertragungsfunktion** ist die Einstellung, mit der Bilder entsprechend den Vorlieben des Benutzers angezeigt werden können. Standardmäßig ist dies auf *Linear* eingestellt. Da dies wirklich das Bild darstellt, wird Anfängern empfohlen, diese Einstellung auf der Standardeinstellung zu belassen. Man vergisst leicht, dass man sich in der automatisch angepassten Ansicht befindet und versteht nicht, warum die gespeicherten Bilder nicht so sind, wie sie auf dem Bildschirm erscheinen. Sie können die Visualisierung im Hauptfenster jedoch jederzeit anpassen.
- Nach dem gleichen Prinzip können die Histogramme mit dem **Standardanzeigemodus** in zwei Modi angezeigt werden. Entweder der *Linear*- oder der *Logarithmische*-Modus. Letzteres kann mit dem Werkzeug „Generalisierte Hyperbolische Streckung“ sehr nützlich sein. Sie können den Modus jedoch in jedem Fenster mit einem Histogramm ändern. In den Einstellungen geht es darum, das Standardverhalten festzulegen.
- **Region of Interest** ist die Einstellung, mit der Sie die Art der Auswahl auswählen können, die Sie auf den ROI anwenden möchten: *Manuell* oder *Automatisch aus Auswahl*. Diese Optionen werden ausführlich im entsprechenden Abschnitt *Einstellung des ROI* beschrieben.
- **Mausverhalten** ermöglicht die Konfiguration des Mausverhaltens. Seit 1.3.0 kann das Verhalten eines doppelten mittleren Klicks konfiguriert werden, um eine voreingestellte Zoomstufe festzulegen. Die voreingestellte Zoomstufe kann entweder auf „Zoom passend“, „Zoom auf 100 % zentriert auf den Mauszeiger“ oder auf Umschalten zwischen „Zoom passend“ und „Zoom auf 100 % zentriert auf den Mauszeiger“ eingestellt werden. Zusammen mit den anderen Verbesserungen bei der Schwenk- und Zoomsteuerung mit der mittleren Maustaste verbessert dies die Einhandbedienbarkeit von Siril.
- Die Schaltfläche **Mauskonfiguration** öffnet den Mauskonfigurationsdialog. Damit können Sie Ihr Mausverhalten für Siril vollständig anpassen, indem Sie das Standardmausverhalten ändern oder Mausfunktionen hinzufügen oder entfernen. Dies kann insbesondere für behinderte Benutzer nützlich sein, um vorhandene Mausfunktionen anderen Tasten zuzuordnen oder um keine Sondertasten zu benötigen. Außerdem kann es für jeden Benutzer mit einer Maus mit vielen Tasten (oder Zweiwege-Scrollrädern) nützlich sein, um Maustasten oder Scrollachsen neue Aktionen (wie Speichern, Rückgängig, Scrollen durch die aktive Sequenz) auf eine Weise zuzuweisen, die bisher nicht möglich war, oder mit Zweiwege-Scrollrädern.

  Sie müssen sich durch eine Erinnerung klicken, dass dieses Tool das Mausverhalten entfernen oder ändern kann, sodass Ihre Einstellungen nicht mehr der Dokumentation entsprechen. Das Tool verfügt über eine gewisse integrierte Plausibilitätsprüfung (es lehnt doppelte Zuweisungen ab, besteht darauf, dass die Hauptmausaktion, die zur Interaktion mit dem Bild verwendet wird, enthalten ist, und prüft auf Konflikte zwischen Einzelklick- und Doppelklickaktionen, die derselben Taste und demselben Modifikator zugewiesen sind). Wenn Sie Ihre Konfiguration wirklich durcheinander bringen, gibt es eine Reset-Schaltfläche, mit der das gesamte Mausverhalten auf die Standardeinstellungen zurückgesetzt wird.

  Die Nummerierung der Maustasten ist nicht immer eindeutig, insbesondere bei Mäusen mit vielen Tasten, da die Richtungen des Scrollrads bei der Nummerierung der Tasten berücksichtigt werden. Daher ist rechts neben dem Dialogfeld ein Klickbereich vorhanden. Wenn Sie hier auf eine beliebige Maustaste klicken, wird deren Nummer in der Beschriftung angezeigt. Dies ist dieselbe Nummer, die Sie verwenden müssen, wenn Sie dieser Maustaste Funktionen zuweisen.


Mouse Action Preferences
Mouse Click Actions
Name Button Action Modifier
Main multifunction action 1 Single click None
Drag 1 Single click Ctrl
Measure 1 Single click Ctrl Shift
Open if nothing is loaded 1 Double click None
Drag 2 Single click None
Photometry box 2 Single click Ctrl
Configurable zoom toggle 2 Double click None
Mouse Menu / Remove 3 Single click None
Add
Remove
Mouse button / scroll check
Scroll Actions
Name Direction Modifier
Scroll to zoom Vertical None
Scroll through sequence Vertical Shift
Add
Remove
Close
Reset
Apply

> **Tipp**
>
> Beachten Sie, dass diese Konfiguration nur für Mausereignisse im Hauptzeichenbereich (wo das Bild angezeigt wird) gilt. Unabhängig von den hier konfigurierten Mausaktionen bleibt das Mausverhalten an anderen Stellen in der Anwendung gleich. Wenn Sie beispielsweise die linke und rechte Maustaste universell vertauschen möchten, können Sie dies hier nicht tun, aber möglicherweise können Sie dies in Ihren Betriebssystemeinstellungen oder in der Maustreiberkonfiguration tun.

- Im Abschnitt **Farben** können Sie die Farben einiger Texte und Symbole konfigurieren, die auf dem Bild angezeigt werden sollen. Klicken Sie dazu einfach auf den Farbbutton und wählen Sie die gewünschte Farbe aus. Diese Auswahl betrifft 5 Punkte:
  - Proben der Hintergrund-Extraktion
  - Standardbeschriftungen
  - Beschriftungen von Deep-Sky-Objekten
  - Beschriftungen von Objekten des Sonnensystems
  - Temporäre Objektbeschriftungen

4.1.8 Farbmanagement

Arbeitsfarbräume

Bevorzugter Farbraum

Diese Option wählt die Arbeitsfarbräume aus, nämlich einen linearen RGB-Arbeitsraum, einen nichtlinearen RGB-Arbeitsraum mit der Standard-Tonwertkurve (TRC) des Farbraums und einen Grau-Arbeitsraum mit einer an den RGB-Arbeitsraum angepassten TRC. Die Optionen sind sRGB, das sich am besten für die Arbeit mit Bildern für die WWW-Nutzung eignet; Rec2020, der von Siril empfohlene Wide-Gamut-Farbraum; und Benutzerdefiniert, mit dem Sie mithilfe der Dateiauswahl Ihre eigenen Profile definieren können.

Externe Farbprofile

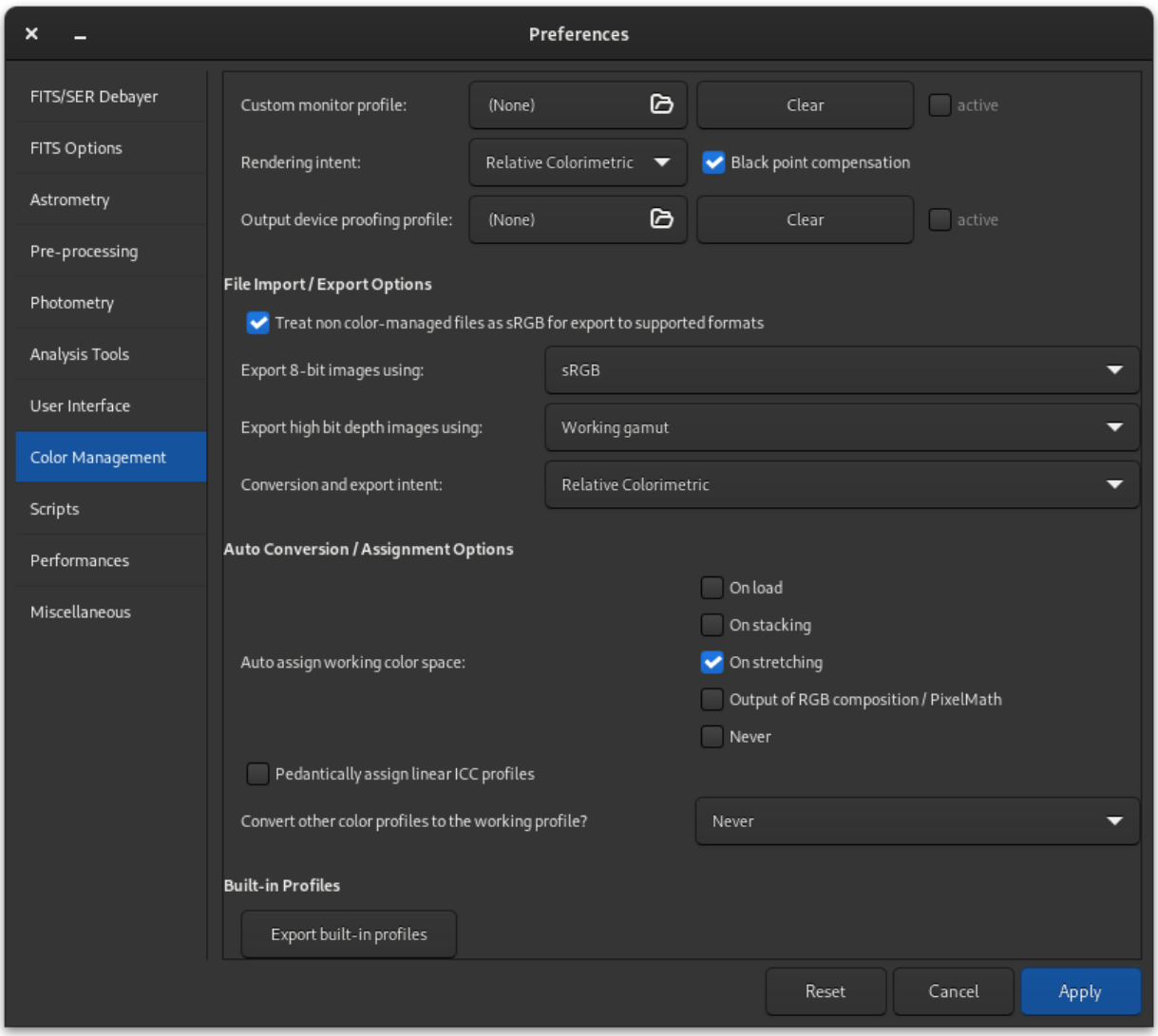
Diese Dateiauswahlmöglichkeiten sind nur in Verbindung mit der oben für Ihren bevorzugten Farbraum ausgewählten Option „Aus Dateien“ von Nutzen.

- **RGB-ICC-Profil**: Mit dieser Dateiauswahl können Sie das RGB-ICC-Profil für Ihren bevorzugten Farbraum auswählen.
- **Graues ICC-Profil**: Mit dieser Dateiauswahl können Sie ein graues ICC-Profil auswählen. Das Profil sollte eine Tone-Response-Kurve (TRC) haben, die mit der des von Ihnen gewählten RGB-Farbraums übereinstimmt.

Ein Beispiel zum Festlegen eines benutzerdefinierten RGB-Farbraums entsprechend Pro Photo RGB finden Sie auf der *Profil-Seite*.

Gamut/Farbraum-Visualisierung

Diese Schaltfläche öffnet ein Referenzbild, das die Farbskalen der integrierten sRGB- und Rec2020-Farbräume mit den Farbskalen einiger anderer bekannter RGB-Farbräume vergleicht.


Preferences
FITS/SER Debayer
FITS Options
Astrometry
Pre-processing
Photometry
Analysis Tools
User Interface
Color Management
Scripts
Performances
Miscellaneous
Custom monitor profile:
(None)
Clear
active
Rendering intent:
Relative Colorimetric
Black point compensation
Output device proofing profile:
(None)
Clear
active
File Import / Export Options
Treat non color-managed files as sRGB for export to supported formats
Export 8-bit images using:
sRGB
Export high bit depth images using:
Working gamut
Conversion and export intent:
Relative Colorimetric
Auto Conversion / Assignment Options
Auto assign working color space:
On load
On stacking
On stretching
Output of RGB composition / PixelMath
Never
Pedantically assign linear ICC profiles
Convert other color profiles to the working profile?
Never
Built-in Profiles
Export built-in profiles
Reset
Cancel
Apply

Benutzer-Farbprofile

- **Benutzerdefiniertes Monitorprofil**: Mit dieser Option können Sie ein benutzerdefiniertes Monitorprofil auswählen, beispielsweise wenn Sie Ihren Monitor kalibriert haben oder einen Monitor mit großem Farbraum verwenden. Außerdem gibt es eine Schaltfläche zum Löschen der aktuellen Auswahl und ein Kontrollkästchen zum Festlegen, ob das benutzerdefinierte Monitorprofil aktiv ist oder nicht. Wenn es nicht aktiv ist, zielt die Anzeigetransformation auf eine Standard-sRGB-Anzeige ab.
- **Rendering-Absicht**: Mit dieser Option können Sie Ihren bevorzugten ICC-Rendering-Absicht auswählen, die für die Farbtransformation vom Arbeitsraum zum Monitorprofil verwendet wird. Eine Beschreibung der verfügbaren Absichten finden Sie *auf der Theorieseite*. Außerdem gibt es eine Check-Schaltfläche, mit der Sie festlegen können, ob die Schwarzpunktkompensation verwendet werden soll oder nicht.
- **Proofing-Profil für Ausgabegeräte**: Mit dieser Option können Sie ein Softproofing-Profil für ein Ausgabegerät auswählen. Wenn hier kein Profil eingestellt ist, richtet sich das Softproofing nach dem Anzeigeprofil. Außerdem gibt es eine Schaltfläche zum Löschen der aktuellen Auswahl und ein Kontrollkästchen zum Festlegen, ob das Softproofing-Profil aktiviert ist oder nicht. Wenn es nicht aktiviert ist, ist der Softproof-Anzeigemodus nicht verfügbar.

Datei Import-/Export-Optionen

- **Nicht farbverwaltete Dateien beim Export in unterstützte Formate als sRGB behandeln**: Diese Option gilt für Bilder, denen kein Farbprofil zugewiesen ist. Wenn die Option aktiviert ist und ein Bild ohne ICC-Profil in ein Format exportiert wird, welches ICC-Profile unterstützt, wird es mit einem sRGB-ICC-Profil oder einem Grauprofil mit einem TRC gespeichert, der dem Standard-sRGB-TRC entspricht. Dies ist im Wesentlichen das Verhalten früherer Versionen von Siril. Es funktioniert oft einwandfrei, stellt jedoch keinen ordnungsgemäß farbverwalteten Arbeitsablauf dar. Die Option ist standardmäßig aktiviert, aber es ist besser, das Problem zu vermeiden, indem Sie die Optionen für die automatische Konvertierung/Zuweisung verwenden, um sicherzustellen, dass Bildern zum Zeitpunkt der Speicherung ein ICC-Profil zugewiesen wird und die richtige Farbverwaltung vorliegt.
- **8-Bit-Bilder exportieren mit**: Diese Option legt den Farbraum für den Export von 8-Bit-Bildern fest, die das Einbetten von ICC-Profilen unterstützen: Sie haben die Wahl zwischen sRGB und Ihrem Arbeitsfarbraum, wie oben mit der Option „Arbeits-RGB-Farbraum“ ausgewählt.
- **Bilder mit hoher Bittiefe exportieren mit**: Diese Option legt den Farbraum für den Export von 16-Bit-Bildern fest, die das Einbetten von ICC-Profilen unterstützen: Sie haben die Wahl zwischen sRGB und Ihrem Arbeitsfarbraum, wie oben mit der Option „Bevorzugter Farbraum“ ausgewählt.
- **Exportabsicht**: Hiermit wird die Absicht für die Umwandlung von Bildern aus dem ihnen zugewiesenen Farbraum in den Farbraum festgelegt, in dem sie gespeichert werden sollen. Behalten Sie die Einstellung „Relativ farbmetrisch“ bei, um sicherzustellen, dass Ihre gespeicherten Bilder genauso aussehen, wie Sie sie gespeichert haben. Andere Absichten sollten hier wahrscheinlich immer nur für Debugging-Zwecke und ähnliches gewählt werden.

Eine Einstellung finden Sie auf der Registerkarte FITS-Optionen:

- **ICC-Profile in FITS-Bilder und FITSEQ-Sequenzen einbetten**: Diese Option ermöglicht die Einbettung von ICC-Profilen in FITS-Dateien: Die Profile werden in eine zusätzliche HDU in der FITS-Datei eingebettet, mit dem Namen „ICCProfile“, die der verwendeten Einbettungsmethode anderer großer Astrofotografie-Software entspricht. *Es wird dringend empfohlen, diese Option aktiviert zu lassen.*

> **Tipp**
>
> Der Export von Bildern in 8-Bit- und 16-Bit-Formate, die keine eingebetteten ICC-Profile unterstützen, erfolgt immer im sRGB-Farbraum.

Automatische Konvertierungs-/Zuweisungsoptionen

Arbeitsfarbraum automatisch zuweisen

Mit diesen Kontrollkästchen legen Sie fest, wann einem Bild automatisch der Arbeitsfarbraum zugewiesen wird. Die Möglichkeiten sind: beim Laden, beim Stacking, beim Stretching oder bei der Ausgabe von Kompositionswerkzeugen (Pixelmath / RGB-Komposition).

Standardmäßig wird der Arbeitsfarbraum beim Stretching zugewiesen, da dies der Punkt ist, an dem Sie normalerweise den Viewer-Modus auf „Linear“ umschalten und beginnen, Ihr Bild so zu strecken, dass es wie gewünscht aussieht. Ab diesem Zeitpunkt beginnt das Farbmanagement eine Rolle zu spielen.

Sie können den Farbraum jedoch auch zu einem früheren Zeitpunkt der Verarbeitung zuweisen, wenn Sie möchten. In den meisten Fällen hat dies nur geringe Vor- oder Nachteile. Wenn Sie jedoch Bilder exportieren möchten, bevor Sie sie strecken, stellen Sie durch Aktivieren der anderen Kontrollkästchen sicher, dass immer ein Farbprofil zugewiesen wird.

Tipp

Ein Farbprofil wird **niemals** automatisch zugewiesen, wenn Siril davon ausgeht, dass es sich um einen einzelnen Kanal handelt, der aus einem 3-Farben-Bild mithilfe von RGB-, HSV-, HSL-, CIE L*a*b*- oder Grün-Extraktion extrahiert wurde. Ein sinnvoll zuordenbares ICC-Profil liegt in diesem Fall nicht vor. Sie sollten die Datei als Rohdaten betrachten, bis Sie bereit sind, sie entweder mit anderen Daten neu zusammenzusetzen oder sie so zu verarbeiten, dass sie für sich genommen gut aussieht. Zu diesem Zeitpunkt sollten Sie Ihren bevorzugten Arbeitsfarbraum zuweisen.

Aufgeteilten CFA-, Ha- und OIII-Kanälen kann problemlos eine lineare Version des Arbeitsraums zugewiesen werden. Dies geschieht, wenn die unten beschriebene pedantische Option aktiv ist.

Wenn Sie Farbprofile vollständig vermeiden möchten, können Sie „Nie“ auswählen. Dies führt jedoch zu einem nicht farbverwalteten Arbeitsablauf und Ihre Bilder sehen möglicherweise nicht konsistent aus, wenn sie in anderer Software angezeigt oder auf anderen Ausgabegeräten ausgegeben werden.

Lineare ICC-Profile pedantisch zuweisen

Beim automatischen Zuweisen von ICC-Profilen zu:

- geladene Bilder, denen ein ICC-Profil fehlt und die keine Anzeichen einer vorherigen Streckung aufweisen,
- neu gestackten Bildern,
- neu zusammengesetzten Bildern

Siril weist ein lineares ICC-Profil zu. Dies ist in der Regel technisch korrekt, da solche Bilder weiterhin lineare Lichtdaten enthalten. Die Option wird zur Verfügung gestellt, wenn Sie sie wünschen, ist jedoch im Allgemeinen nicht erforderlich: Eine Erklärung dazu finden Sie auf der *Profile-Seite*. Möglicherweise möchten Sie dies verwenden, um von der Anzeigetransformation zu profitieren, indem Sie Ihre Streckungen auf die ursprünglichen linearen Daten anwenden und gleichzeitig den Effekt in sRGB (oder Ihrem benutzerdefinierten Monitorprofil) in der Vorschau anzeigen. In diesem Fall sollten Sie das folgende Kontrollkästchen für die automatische Konvertierung deaktivieren, damit das Profil während des Streckens linear bleibt, und konvertieren Sie die Daten nach Abschluss des Dehnens manuell in Ihr Arbeitsprofil.

Andere Farbprofile in Arbeitsfarbraum konvertieren?

Dieses Dropdown-Menü legt das Verhalten von Siril fest, wenn ein Bild mit einem ICC-Profil geladen wird, das sich vom ausgewählten Arbeitsprofil unterscheidet. Die Optionen sind: Nie, Immer und Immer fragen. Die Standardeinstellung ist „Nie“, da dadurch unaufgeforderte Änderungen am Bild vermieden werden. Für einen konsistenten Arbeitsablauf können Sie jedoch „Immer“ auswählen, damit Sie immer in Ihrem bevorzugten Farbraum arbeiten. „Jedes Mal nachfragen“ ist eine Option, die jedoch nervig sein kann, insbesondere weil die Exportprofile der Version 2 als anders angesehen werden als die integrierten Arbeitsprofile der Version 4. Obwohl sie für Benutzer bereitgestellt wird, die jedes Detail verwalten möchten, wird diese Option nicht unbedingt empfohlen.

Eingebaute Profile

Eingebaute Skripte exportieren

Siril verwendet Profile aus der Elle Stone-Farbprofilkollektion, um seine integrierten Profile bereitzustellen. Da diese unter einer Creative Commons BY-SA-3.0-Lizenz vertrieben werden, die die Freigabe der Profile erfordert, steht eine Schaltfläche zum Exportieren aller integrierten Profile in das Arbeitsverzeichnis zur Verfügung.

Um ein Profil bereitzustellen, das die Render-Absicht „Wahrnehmungsbezogener sRGB“ unterstützt, wird das ICC-Präferenzprofil mit wahrnehmungsbezogenen Nachschlagetabellen bereitgestellt. Um dies zu exportieren, müssen Sie den gleichen Allgemeinen Geschäftsbedingungen zustimmen wie beim Herunterladen von der ICC-Website.

4.1.9 Skripte

- Die Registerkarte Skripte enthält im Wesentlichen die Pfadangaben, an denen Siril nach Skripten suchen soll. Tatsächlich werden die Skripte standardmäßig und je nach verwendetem Betriebssystem an einem bestimmten Ort installiert:
  - `/usr/local/share/siril/scripts` oder `/usr/share/siril/scripts` unter GNU/Linux.
  - `C:\Program Files\Siril\scripts` in Windows.
  - `/Applications/Siril.app/Contents/Resources/share/siril/scripts` in MacOS, wenn die Anwendung im Anwendungsordner installiert wurde.

> **Warnung**
>
> Da die Anwendung unter macOS signiert und beglaubigt ist, ist es unmöglich, die Skripte innerhalb des Programmverzeichnisse zu ändern. Andernfalls wird die Anwendung nicht gestartet. Sie müssen also einen anderen Pfad definieren, der auf einen Ordner verweist, für den Sie Schreibrechte haben.

- Im Feld **Speicherort für Skripte** können Sie benutzerdefinierte Ordnerpfade definieren, in denen von Ihnen erstellte und/oder geänderte Skripte abgelegt werden. Wenn Sie auf die Schaltfläche darunter klicken, werden die Ordner erneut durchsucht und die Liste der Skripte im entsprechenden Menü aktualisiert.
- Im Abschnitt **Warnungdialoge** wird vorgeschlagen, sie zu deaktivieren:
  1. Der Warntext, der angezeigt wird, bevor ein Skript ausgeführt wird.
  2. Die Überprüfung des Schlüsselwortes `requires` (Benötigt), das ganz am Anfang des Skripts stehen muss, um zu prüfen, ob das Skript mit der Version von Siril kompatibel ist. Wir empfehlen, diese Option nicht zu deaktivieren.
- The **Siril Script Repository** field is where scripts that have been validated by the Siril team are automatically uploaded. However, these may have been written by third-party developers. You can enable/disable this feature with the *Enable use of the siril-scripts online repository* button, and define automatic or manual updates with *Automatic updates* and *Manual update*. To display a script in the script menu, simply select it.

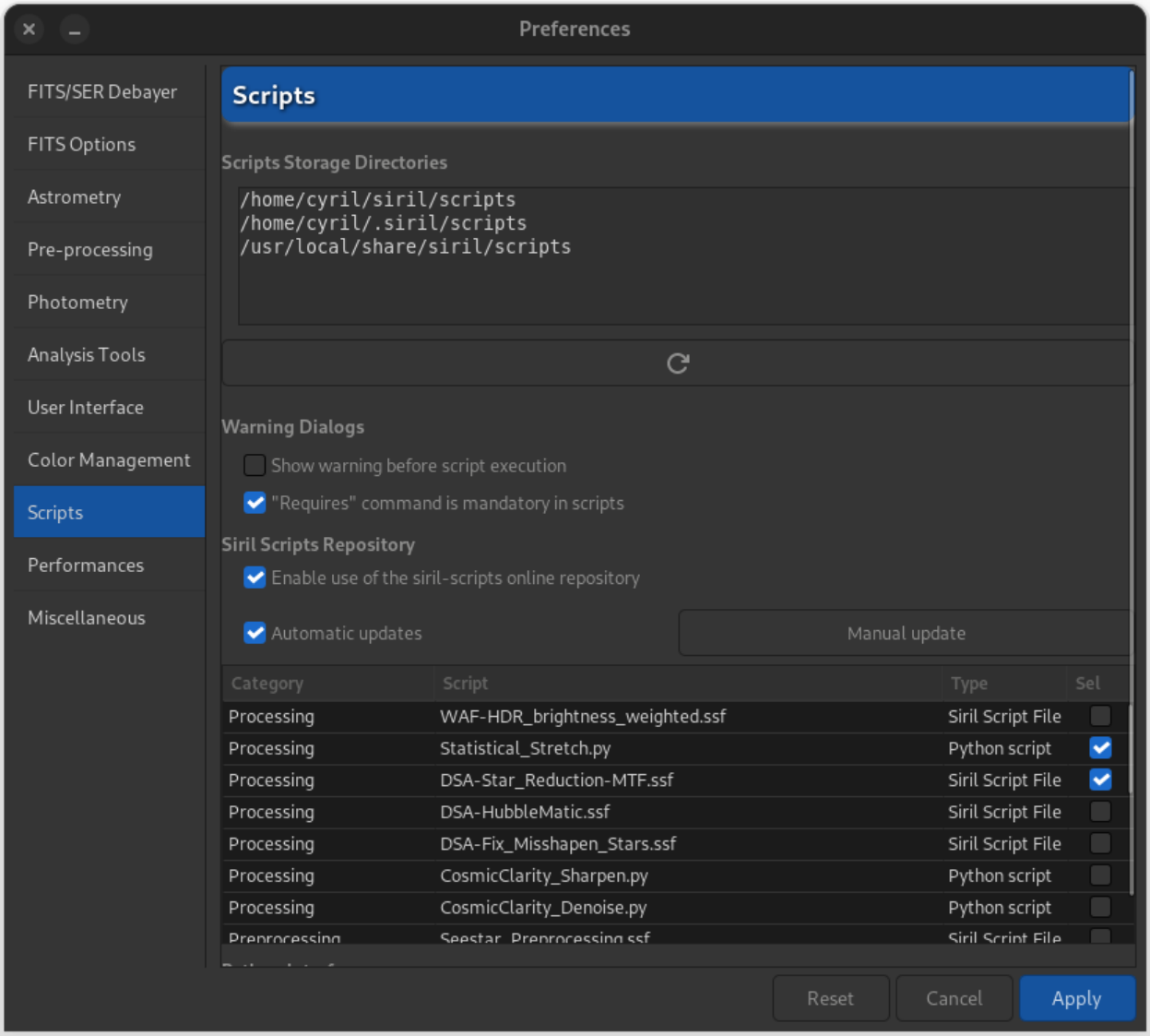


Abb. 12: Seite 8 des Einstellungs-Dialogs

- The **Reset python venv** will kill all running python scripts and reset the virtual environment "venv" directory to an as-new state. Any modules that have been installed by scripts will require re-installation. This is normally not necessary but is provided as a last resort to restore the venv to a pristine state as an aid to troubleshooting.

4.1.10 Leistungseinstellungen (Performance)

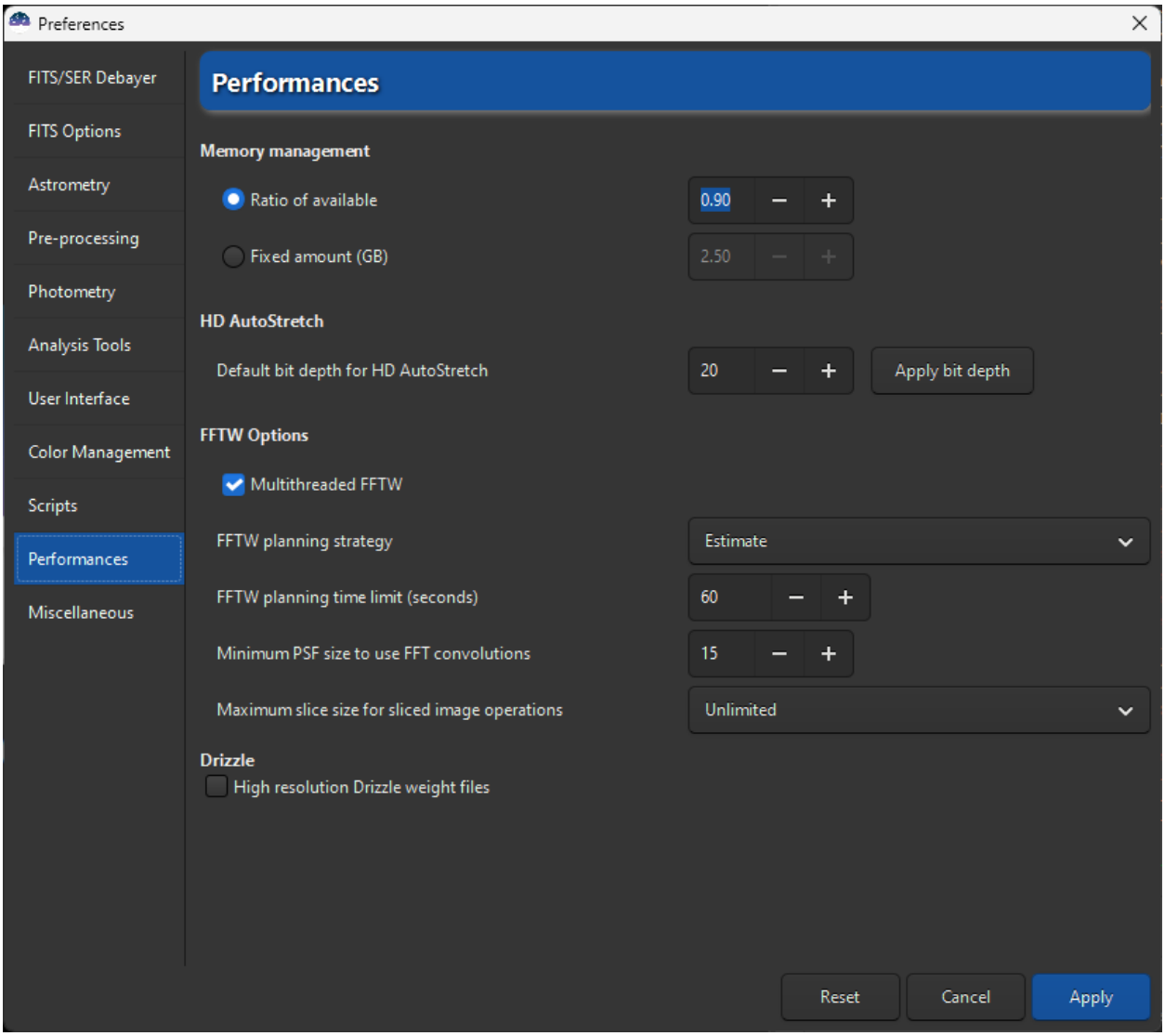


Abb. 13: Seite 9 des Einstellungs-Dialogs

Astronomische Bildverarbeitungssoftware wie Siril verbraucht viele Ressourcen und erfordert in der Regel recht leistungsstarke Computer. Wenn der Computer stark ausgelastet ist, ist es nicht ausgeschlossen, dass er komplett einfriert. Es ist nicht empfehlenswert, während der Verarbeitung irgendetwas anderes auf dem Computer zu tun, vor allem nicht im Internet zu surfen, da die Browser sehr viel Arbeitsspeicher benötigen. Es ist jedoch möglich, den maximalen Prozentsatz des Arbeitsspeichers zu verwalten, den Siril verwenden kann.

- **Nicht verfügbar**: Siril beschränkt sich auf einen Anteil der Menge des freien physischen Hauptspeichers und verringert bei Bedarf die Größe der Arbeitsprozesse. Ein Wert über 1 bedeutet, dass ein Teil des Speichers, der in eine Datei/Festplatte ausgelagert wird, verwendet wird und dass der Gesamtprozess langsamer wird und das System während einiger Operationen wahrscheinlich nicht reagiert. Wenn Sie keinen Auslagerungsspeicher konfiguriert haben, führt ein Wert von 1 oder höher wahrscheinlich zu einem Absturz von Siril oder des Betriebssystems.

- **Fester Wert (GB)**: Siril beschränkt sich selbst auf eine feste Speichermenge und verringert bei Bedarf die Größe der Arbeitsprozesse. Wenn Sie eine größere Speichermenge konfigurieren, als auf Ihrem System verfügbar ist, kann dies zu einem Absturz von Siril oder des Betriebssystems führen.
- Die Option **Standard-Bittiefe für HD AutoStretch** legt die Standard-Bittiefe für den HD AutoStretch-Anzeigemodus fest. Höhere Bittiefen erfordern exponentiell mehr Speicher für die Farbtabelle (LUT) und benötigen mehr Zeit für die Neuberechnung, aber sie glätten Quantisierungsartefakte bei der Anzeige von Bildern mit sehr schmalen Histogrammspitzen wesentlich besser. Die Standard-Bittiefe gilt ab dem nächsten Wechsel des Ansichtsmodus und kann jetzt über die Schaltfläche auf der rechten Seite eingestellt werden. Klicken Sie auf die Schaltfläche *Bit-Tiefe anwenden*, um die ausgewählte HD AutoStretch-Bittiefe jetzt einzustellen.
- **Multithreaded FFTW**: Mit dieser Schaltfläche wird FFTW so eingestellt, dass mehrere Threads verwendet werden. Dies kann schneller sein (obwohl die Leistung aufgrund des Synchronisations-Overheads nicht linear mit der Anzahl der Prozessoren ansteigt), aber die Planungsphase von FFTW dauert bei Multi-Thread-Systemen länger, so dass insbesondere die erste FFT für eine bestimmte Bildgröße bei Verwendung mehrerer Threads erheblich langsamer sein kann.
- **FFTW-Planungsstrategie**: Diese Combobox legt die FFTW-Planungsstrategie fest. FFTW verfügt über mehrere Algorithmen zur Berechnung einer FFT und plant eine gegebene FFT, um die Geschwindigkeit zu optimieren. Die Ergebnisse dieser Planungen werden für die spätere Wiederverwendung in einer Cachedatei namens "Wisdom" (Weisheit) gespeichert, so dass sich ein gewisser zusätzlicher Zeitaufwand für die Planung im Vorfeld lohnen kann, wenn Sie viele FFTs der gleichen Größe berechnen. Beachten Sie, dass die Weisheit für einen bestimmten Rechner spezifisch ist: Sie sollte nicht zwischen verschiedenen Computern ausgetauscht werden und nach einer Speicher- oder Prozessorerweiterung oder einer größeren Änderung der Softwareumgebung (größere Änderungen des Betriebssystems, größere Versionsänderungen von Siril) gelöscht und von Grund auf neu erstellt werden. In der Rangfolge der Arbeitsgeschwindigkeiten ist Schätzen am schnellsten: Bei dieser Strategie werden keine Messungen durchgeführt, sondern eine Planung auf der Grundlage einer Reihe von Heuristiken vorgenommen. Messen ist die Nächstschnellste: Diese Methode vergleicht die Geschwindigkeit der verschiedenen internen FFTW-Methoden zur Berechnung der FFT und wählt die schnellste aus. Infolgedessen dauert der Planungsschritt länger. Patient berücksichtigt noch mehr mögliche Pläne, und Exhaustive noch mehr. Wenn Sie immer Bilder einer bestimmten Größe verarbeiten, können sich die komplexen Planungsstrategien aufgrund der Weisheit lohnen, aber wenn Sie mit Bildern vieler verschiedener Größen arbeiten, kann ein einfacherer Planungsalgorithmus besser geeignet sein.
- **FFTW-Planungszeitlimit**: Mit diesem Zeitlimit wird die FFTW-Planung nach dem angegebenen Zeitlimit angehalten. Damit wird die Planungsstrategie außer Kraft gesetzt. Beachten Sie, dass das Zeitlimit nicht streng erzwungen wird: FFTW beendet alle nicht unterbrechbaren Berechnungen, die es zum Zeitpunkt des Erreichens des Limits durchführt, und wenn es auf Null gesetzt wird, führt FFTW zumindest immer eine Schätzung der Planung durch.
- **Drizzle: High resolution Drizzle weight files**: When this button is toggled, weight files produced by the drizzle algorithm will be of the same bitdepth than the images. This can be important when trying to get the best possible FWHM on the stacked image, but it is very much space consuming. By default, it is unticked and the weight files are saved in 8-bit.

4.1.11 Verschiedenes

Die letzte Registerkarte enthält alles, was nicht in die anderen Themen passt.

- Die Verwendung der Schaltflächen *Undo*/*Redo* erfordert Speicherplatz. In manchen Fällen sehr viel Platz. Der Ordner, der die Auslagerungsdateien enthält (das sind die Dateien, die für das ordnungsgemäße Funktionieren der Undo/Redo-Funktionen erforderlich sind), kann im Abschnitt **Verzeichnis für Auslagerungsspeicher** festgelegt werden. Der Speicherplatz wird rechts neben der Dateiauswahl angezeigt. Wir raten **dazu, die Standardeinstellungen nicht zu ändern**, es sei denn, Sie haben einen triftigen Grund, dies zu tun. Da die Wahl eines guten Ordners kritisch ist, ist es möglich, zum Standardordner zurückzukehren, indem Sie auf *Standard wiederherstellen* klicken.
- Die **Warnungsdialoge** ermöglichen es, einige Warn-Popups zu deaktivieren, die Anfängern helfen sollen.

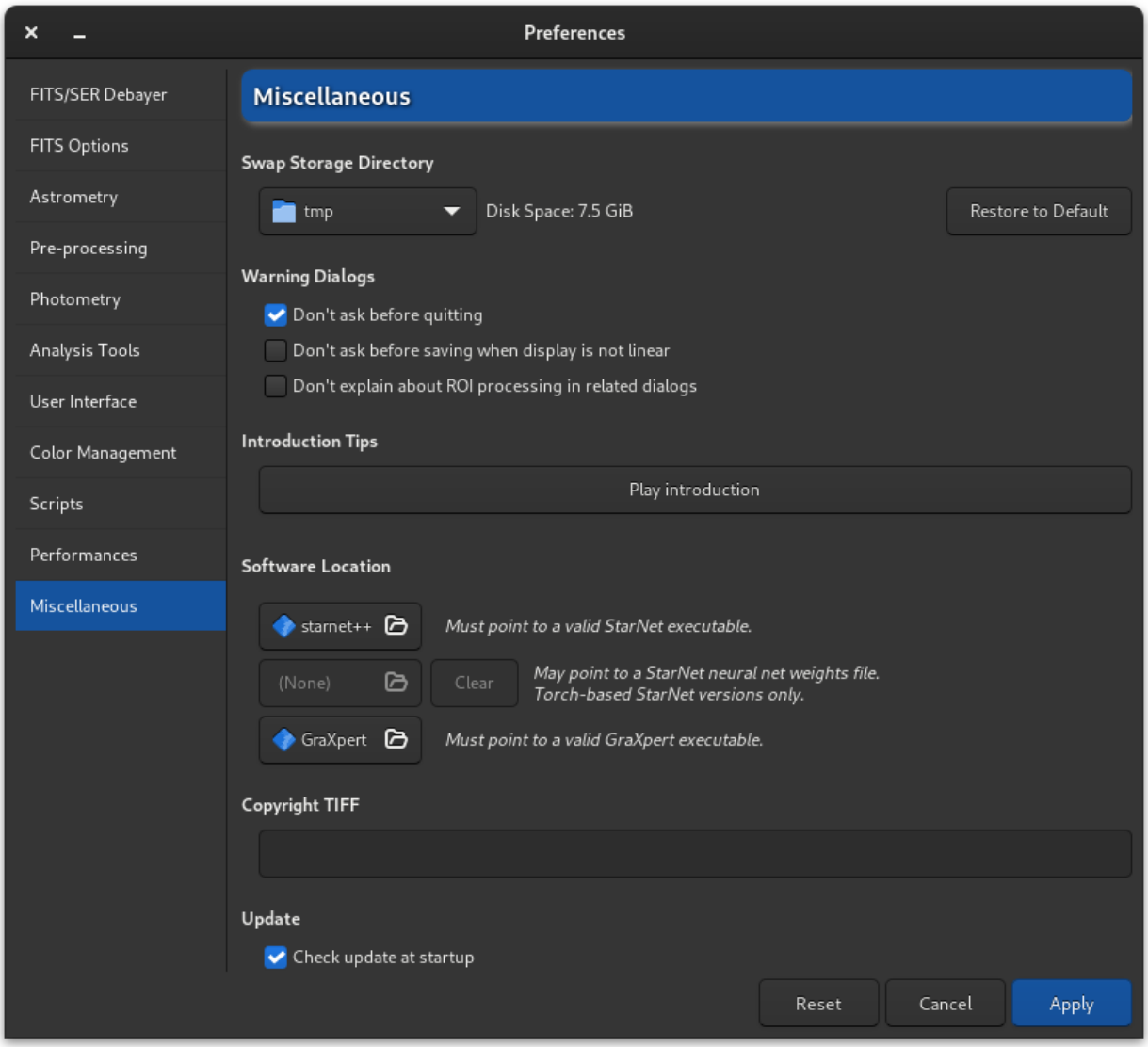


Abb. 14: Seite 10 des Einstellungs-Dialogs

- **Einführungstipps**: Beim ersten Start von Siril können Sie eine kleine Animation sehen, die zeigt, was es Neues in der Anwendung gibt. Diese Animation kann durch Klicken auf *Einführung abspielen* abgespielt werden.
- **Speicherort der ausführbaren StarNet-Datei**: Um StarNet in Siril verwenden zu können, muss Siril der Pfad mitgeteilt werden, in dem sich die ausführbare StarNet-Datei befindet. Bei alten StarNet++ v1-Installationen, die getrennte ausführbare Dateien für die Verarbeitung von Mono- und RGB-Dateien verwenden, kann eine von beiden gewählt werden - Siril erkennt automatisch die andere, wenn beide installiert sind. Beachten Sie, dass für diese alten Installationen die ursprünglichen Namen der ausführbaren Dateien **rgb_starnet++** und **mono_starnet++** beibehalten werden MÜSSEN. Bei allen neueren einzeln ausführbaren Versionen von StarNet ermittelt Siril die Version heuristisch und stellt eine entsprechende Schnittstelle her.
- **Speicherort der StarNet-Gewichte**: Neue Torch-basierte Versionen von StarNet bieten die Möglichkeit, den Speicherort einer Gewichtungsdatei für ein neuronales Netz anzugeben: Sie muss sich nicht im selben Verzeichnis wie die ausführbare Datei befinden. Diese Einstellung kann verwendet werden, um den Speicherort einer Gewichtungsdatei festzulegen, die an StarNet++ übergeben werden soll, und sie kann über die zugehörige Schaltfläche zurückgesetzt werden. Hinweis: Diese Option funktioniert nur bei Torch-basierten StarNet-Installationen. Bei älteren StarNet-Installationen muss sich die Gewichtungsdatei im selben Verzeichnis wie die ausführbare Datei befinden.

> **Warnung**
>
> Dies ist das Verzeichnis der Kommandozeilenversion von Starnet++, das angegeben werden muss, nicht das der GUI-Version.

> **Warnung**
>
> Unter macOS kann es schwierig sein, den Verzeichnispfad zu finden, da Apple das Durchsuchen einiger Ordner nicht einfach macht. Ein Trick ist es, `Umschalt` + `Cmd` + `g` in den geöffneten Dateiauswahldialog einzugeben, und dann direkt den Installationspfad einzugeben, der normalerweise der von Hombrew eingestellte ist. Normalerweise ist dies `/usr/local/bin` auf Intel-Computern und `/opt/homebrew/bin/` auf Apple Silicon Versionen.

- **Speicherort der ausführbaren GraXpert-Datei**: Um GraXpert in Siril verwenden zu können, muss Siril der Pfad mitgeteilt werden, in dem sich die ausführbare GraXpert-Datei befindet. Wenn Sie Schwierigkeiten haben, GraXpert mit Siril zum Laufen zu bringen, prüfen Sie, ob es von selbst funktioniert, bevor Sie einen Fehlerbericht senden: Siril führt lediglich die Drittanbietersoftware aus. Wenn es also nicht von selbst funktioniert, funktioniert es auch nicht, wenn es von Siril aufgerufen wird.
- **Copyright TIFF**: Beim Speichern von TIFF-Dateien ist es möglich, das Copyright der zugehörigen EXIF-Metadaten anzupassen.
- **Aktualisierung**: Standardmäßig prüft Siril beim Starten ob Updates verfügbar sind. Es steht Ihnen frei, dieses Verhalten zu deaktivieren, wenn Sie nicht möchten, dass die Anwendung unsere Website abfragt.

4.2 Einstellungen (Befehle)

Ab Version 1.2 können die meisten Einstellungen auch durch Befehle gesetzt werden, d.h. entweder durch direkte Eingabe in der Kommandozeile, durch Skripte oder im Headless-Modus.

Um eine Liste aller verfügbaren Variablen zu erhalten, verwenden Sie die Befehlszeile von Siril:

```
get -A
```

Dies gibt eine Liste aller Variablen auf der Konsole aus, mit ihrem aktuellen Wert und einer kurzen Beschreibung (verwenden Sie die Kleinbuchstaben-Option *-a*, um die Beschreibung wegzulassen).

In der nachstehenden Tabelle sind sie aufgeführt:

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
core.wd	(not set)	Ver-zeich-nispfad	Aktuelles Arbeitsverzeichnis
core.extension	.fit	Zei-chenket-te	FITS Dateiendung
core.force_16bit	false	Boolean	Keine 32-Bit Farbtiefe benutzen
core.fits_save_icc	true	Boolean	ICC-Profile beim Speichern in FITS einbetten
co-re.allow_heterogeneous_fitseq	false	Boolean	FITS-Würfel können unterschiedliche Größen ha-ben
core.mem_mode	0 [0, 1]	Ganz-zahl	Hauptspeicher-Modus (0 Prozent, 1 GB)
core.mem_ratio	0.9 [0.05, 4]	Double	Anteil des nutzbaren Hauptspeichers
core.mem_amount	10 [0.1, 1e+06]	Double	Anteil des nutzbaren Hauptspeichers in GB
core.hd_bitdepth	20 [17, 24]	Ganz-zahl	HD Auto-Stretch Bittiefe
core.script_check_requires	true	Boolean	Benötigt den Befehl requires in Skripten
core.pipe_check_requires	false	Boolean	Benötigt den Befehl requires in Pipes
core.check_updates	false	Boolean	Prüfe beim Start auf Updates
core.lang	(not set)	Zei-chenket-te	Aktivierte GUI-Sprache
core.swap_dir	Abhängig vom Betriebs-system	Ver-zeich-nispfad	Auslagerungsverzeichnis
core.binning_update	true	Boolean	Pixelgröße gebinnter Bilder aktualisieren
core.wcs_formalism	1 [0, 1]	Ganz-zahl	WCS-Formalismus im FITS-Header
core.catalogue_namedstars	~/.lo-cal/share/kstars/	Zei-chenket-te	Pfadname des namedstars.dat Katalogs
core.catalogue_unnamedstars	~/.lo-cal/share/kstars/	Zei-chenket-te	Pfadname des unnamedstars.dat Katalogs
core.catalogue_tycho2	~/.lo-cal/share/kstars/	Zei-chenket-te	Pfadname des deepstars.dat Katalogs
core.catalogue_nomad	~/.local/share/ks NOMAD-1e8.dat	Zei-chenket-te	Pfadname des USNO-NOMAD-1e8.dat Katalogs
core.catalogue_gaia_astro	~/.lo-cal/share/siril/ga	Zei-chenket-te	Pfad des lokalen Gaia-Astrometriekatalogs
core.catalogue_gaia_photo	~/.lo-cal/share/siril/ga	Zei-chenket-te	Pfad des lokalen Gaia-Photometriekatalogs

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
core.rgb_aladin	false	Boolean	CTYPE3='RGB' in den FITS-Header einfügen
core.use_checksum	false	Boolean	Prüfe Dateiprüfsumme wenn vorhanden
core.copyright	(not set)	Zei-chenket-te	Copyrightangabe des Benutzers, das in den Datei-header geschrieben wird
core.starnet_exe	(not set)	Zei-chenket-te	Speicherort der ausführbaren StarNet-Datei
core.starnet_weights	(not set)	Zei-chenket-te	Speicherort der StarNet-Torch Gewichtungsdatei
core.graxpert_path	(not set)	Zei-chenket-te	Speicherort der ausführbaren GraXpert-Datei
core.asnet_dir	(not set)	Zei-chenket-te	Verzeichnis der Installation von asnet_ansvr
core.fftw_timelimit	60	Double	Zeitlimit für die Vorberechnung der FFTW
core.fftw_conv_fft_cutoff	15	Ganz-zahl	Minimale PSF-Größe für FFT-Faltungen
core.fftwf_strategy	0	Ganz-zahl	FFTW planning strategy
core.fftw_multithreaded	true	Boolean	FFTW mit parallelen Threads berechnen
core.max_slice_size	32769 [512, 32769]	Ganz-zahl	Maximum slice size for automated slice proces-sing
starfinder.focal_length	0 [0, 999999]	Double	Brennweite in mm für die Einstellung des PSF-Radius
starfinder.pixel_size	0 [0, 99]	Double	Pixelgröße des Sensors in µm für die Radiusbe-rechnung
debayer.use_bayer_header	true	Boolean	Bayer-Muster aus dem Header verwenden
debayer.pattern	0 [0, 7]	Ganz-zahl	Index des Bayer-Musters
debayer.interpolation	8 [0, 9]	Ganz-zahl	Art der Interpolation
debayer.orientation	0 [0, 3]	Ganz-zahl	row-order preference
debayer.offset_x	0 [0, 1]	Ganz-zahl	Bayer Matrix Offset x
debayer.offset_y	0 [0, 1]	Ganz-zahl	Bayer Matrix Offset Y
debayer.xtrans_passes	1 [1, 4]	Ganz-zahl	Anzahl der Durchläufe für den X-Trans Markesteijn-Algorithmus
photometry.gain	2.3 [0, 10]	Double	Elektronen pro ADU für die Schätzung des Rau-schens
photometry.inner	20 [2, 100]	Double	innerer Radius für den Hintergrundring
photometry.outer	30 [3, 200]	Double	Außenradius für den Hintergrundring
photometry.inner_factor	4.2 [2, 50]	Double	Faktor für die automatische Berechnung des In-nenradius

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
photometry.outer_factor	6.3 [2, 50]	Double	Faktor für die automatische Berechnung des Außenradius
photometry.force_radius	true	Boolean	erzwingt Flux-Aperturwert
photome-try.auto_aperture_factor	4 [1, 5]	Double	Radius/Halb-FWHM Verhältnis
photometry.aperture	10 [1, 100]	Double	erzwungene Blende für die Flussberechnung
photometry.minval	-1500 [-65536, 65534]	Double	MInimaler erlaubter Pixelwert für Photometrie
photometry.maxval	60000 [1, 65535]	Double	Maximaler erlaubter Pixelwert für Photometrie
photome-try.discard_var_catalogues	4 [0, 7]	Ganz-zahl	Kataloge, die verwendet werden sollen, um die variablen Sterne aus der Liste der Vergleichssterne zu entfernen
photometry.redpref	(not set)	Zei-chenket-te	Bevorzugtes SPCC Rotfilter
photometry.greenpref	(not set)	Zei-chenket-te	Bevorzugtes SPCC Grünfilter
photometry.bluepref	(not set)	Zei-chenket-te	Bevorzugtes SPCC Blaufilter
photometry.lpfpref	(not set)	Zei-chenket-te	Bevorzugtes SPCC DSLR LPF-Filter
photometry.oscfilterpref	Kein Filter	Zei-chenket-te	Bevorzugtes SPCC OSC-Filter
photometry.monosensorpref	(not set)	Zei-chenket-te	Bevorzugter SPCC Mono-Sensor
photometry.oscsensorpref	(not set)	Zei-chenket-te	Bevorzugter SPCC OSC(Farb)-Sensor
photometry.is_mono	true	Boolean	Ist der SPCC-Sensor monochrom?
photometry.is_dslr	false	Boolean	Ist der SPCC-Farbsensor eine DSLR?
photometry.nb_mode	false	Boolean	sind wir im Schmalband-Modus?
photometry.r_wl	656.28	Double	Wellenlänge des Schmalband-Rotfilters
photometry.r_bw	6	Double	Durchlassbreite des Schmalband-Rotfilters
photometry.g_wl	500.7	Double	Wellenlänge des Schmalband-Grünfilters
photometry.g_bw	6	Double	Durchlassbreite des Schmalband-Grünfilters
photometry.b_wl	500.7	Double	Wellenlänge des Schmalband Blaufilters
photometry.b_bw	6	Double	Durchlassbreite des Schmalband-Blaufilters
astrometry.asnet_keep_xyls	false	Boolean	.xyls Fits-Tabellen nicht löschen
astrometry.asnet_keep_wcs	false	Boolean	.wcs-Ergebnisdateien nicht löschen
astrometry.asnet_show_output	false	Boolean	Ausgabe von solve-field im Hauptprotokoll anzeigen

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
astrometry.sip_order	3 [1, 5]	Ganz-zahl	Grad der Polynomkorrektur
astrometry.radius	10 [0.01, 30]	Double	Radius um die Zielkoordinaten (Grad
astrometry.max_seconds_run	30 [0, 100000]	Ganz-zahl	Maximalzeit in Sekunden für den Lösungsversuch
astrometry.update_default_scale	true	Boolean	Aktualisierung der Standardbrennweite und Pixel-größe anhand des Ergebnisses
astrometry.percent_scale_range	20 [1, 50]	Ganz-zahl	Erlaubter Prozentsatz unterhalb und oberhalb der erwarteten Probenahme
astrometry.default_obscode	(not set)	Zei-chenket-te	Standard IAU Observatoriumscode
analysis.panel	256 [127, 1024]	Ganz-zahl	Panelgröße der Aberrationsprüfung
analysis.window	381 [300, 1600]	Ganz-zahl	Fenstergröße des Aberration Inspektors
compression.enabled	false	Boolean	FITS Komprimierung aktiviert
compression.method	0 [0, 3]	Ganz-zahl	FITS Kompressionsmethode
compression.quantization	16 [8, 256]	Double	Quantisierungsfaktor für 32-Bit-Fließkomma
compression.hcompress_scale	4 [0, 256]	Double	Hcompress Skalierungsfaktor
gui_prepro.cfa	false	Boolean	Art des Sensors für kosmetische Korrekturen
gui_prepro.equalize_cfa	true	Boolean	Flat Kanäle ausgleichen
gui_prepro.fix_xtrans	false	Boolean	aktiviert Korrektur für X-Trans-Sensor
gui_prepro.xtrans_af_x	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_af_y	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_af_w	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_af_h	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_sample_x	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_sample_y	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_sample_w	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.xtrans_sample_h	0	Ganz-zahl	Falls kein X-Trans-Modell gefunden wurde, benut-ze diesen Wert
gui_prepro.bias_lib	(not set)	Zei-chenket-te	Standard Master-Bias
gui_prepro.use_bias_lib	false	Boolean	Benutzer das Standard Master-Bias
gui_prepro.dark_lib	(not set)	Zei-chenket-te	Standard Masterdark
gui_prepro.use_dark_lib	false	Boolean	Benutzer das Standard Masterdark

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
gui_prepro.flat_lib	(not set)	Zei-chenket-te	Standard Master-Flat
gui_prepro.use_flat_lib	false	Boolean	Benutze das Standard Masterflat
gui_prepro.disto_lib	(not set)	Zei-chenket-te	Standard Verzerrungsmuster
gui_prepro.use_disto_lib	false	Boolean	Standard-Master-Verzerrung verwenden
gui_prepro.stack_default	$seqna-me$stacked	Zei-chenket-te	Standardname für das Stacking-Ergebnis
gui_prepro.use_stack_default	true	Boolean	Bevorzugten Stacking-Namen verwenden
gui_registration.method	0 [0, 7]	Ganz-zahl	Index der gewählten Registrierungsmethode
gui_registration.interpolation	4 [0, 5]	Ganz-zahl	Index der gewählten Interpolationsmethode
gui_registration.clamping	true	Boolean	benutzte Begrenzungsmethode mit Lanczos und kubischer Interpolation
gui_registration.drizz_weight_ma	false	Boolean	Match Drizzle weights bitpix to sequence
gui_stack.method	0 [0, 4]	Ganz-zahl	Index der ausgewählten Methode
gui_stack.normalization	3 [0, 4]	Ganz-zahl	Index der Normalisierungsmethode
gui_stack.rejection	5 [0, 7]	Ganz-zahl	Index des Ausschlussverfahrens
gui_stack.weighting	0 [0, 4]	Ganz-zahl	Index der Gewichtungsmethode
gui_stack.sigma_low	3 [0, 20]	Double	Unterer Sigma-Wert für die Ausreißer
gui_stack.sigma_high	3 [0, 20]	Double	Oberer Sigma-Wert für die Ausreißer
gui_stack.linear_low	5 [0, 20]	Double	Unterer linearer Wert für die Ausreißer
gui_stack.linear_high	5 [0, 20]	Double	Oberer linearer Wert für die Ausreißer
gui_stack.percentile_low	3 [0, 100]	Double	Unterer Prozentwert für die Ausreißer
gui_stack.percentile_high	3 [0, 100]	Double	Oberer Prozentwert für die Ausreißer
gui.first_start	(not set)	Zei-chenket-te	Erster Start von Siril
gui.silent_quit	false	Boolean	Beenden nicht bestätigen
gui.silent_linear	false	Boolean	Speichern nicht bestätigen, wenn der Modus nicht linear ist
gui.remember_windows	true	Boolean	Fensterposition merken
gui.main_win_pos_x	0	Ganz-zahl	Position des Hauptfensters
gui.main_win_pos_y	0	Ganz-zahl	Position des Hauptfensters
gui.main_win_pos_w	0	Ganz-zahl	Position des Hauptfensters
gui.main_win_pos_h	0	Ganz-zahl	Position des Hauptfensters

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
gui.pan_position	-1	Ganz-zahl	Position des Trenners zwischen den Fensterteilen
gui.extended	true	Boolean	Hauptfenster ist erweitert
gui.maximized	false	Boolean	Hauptfenster ist maximiert
gui.theme	0 [0, 1]	Ganz-zahl	Index des ausgewählten Themas
gui.font_scale	100	Double	Schriftgröße in Prozent
gui.icon_symbolic	false	Boolean	Stil der Bildschirmsymbole
gui.script_path		Liste von Zei-chenket-ten	Liste der Skriptverzeichnisse
gui.use_scripts_repository	true	Boolean	benutze und synchronisiere das Online Skript-Repository
gui.use_spcc_repository	true	Boolean	benutze und synchronisiere das SPCC-Datenbank-Repository
gui.auto_update_scripts	true	Boolean	Automatische Synchronisierung des Online Skript-Repositories
gui.auto_update_spcc	true	Boolean	Automatische Synchronisierung des SPCC-Datenbank_Repositories
gui.selected_scripts		Liste von Zei-chenket-ten	Liste der aus dem Repository ausgewählten Skripte
gui.warn_scripts_run	true	Boolean	Warnen, wenn ein Skript gestartet wird
gui.show_thumbnails	true	Boolean	Vorschaubilder im Dialog "Datei öffnen" anzeigen
gui.thumbnail_size	256	Ganz-zahl	Größe der Miniaturbilder
gui.selection_guides	0	Ganz-zahl	Anzahl der Elemente der Gitterlinien
gui.show_deciasec	false	Boolean	Zehntel-Bogensekunden anzeigen, wenn der Mauszeiger darüber schwebt
gui.default_rendering_mode	0 [0, 6]	Ganz-zahl	Standard Anzeigemodus
gui.display_histogram_mode	0 [0, 1]	Ganz-zahl	Standard Histogrammanzeige-Modus
gui.roi_mode	0	Ganz-zahl	ROI Auswahlmodus
gui.roi_warning	true	Boolean	ROI Warndialog aktivieren
gui.mmb_zoom_action	0	Ganz-zahl	Zoom-Funktion der mittleren Maustaste
gui.mouse_speed_limit	0	Double	Mouse smooth scroll speed limit
gui.color_bkg_samples	rgba(255, 51, 26, 1.0)	Zei-chenket-te	Konfigurieration der Farbe der Hintergrundsamples
gui.color_std_annotations	rgba(128, 255, 77, 0.9)	Zei-chenket-te	Konfiguration der Standard-Anmerkungsfarbe

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
gui.color_dso_annotations	rgba(255, 128, 0, 0.9)	Zei-chenket-te	Konfiguration Farbe DeepSky-Objektbezeichnungen
gui.color_sso_annotations	rgba(255, 255, 0, 0.9)	Zei-chenket-te	Konfiguration Farbe der Bezeichnung von Objek-ten des Sonnensystems
gui.color_tmp_annotations	rgba(255, 0, 0, 0.9)	Zei-chenket-te	Konfiguration Farbe temporärer Bezeichnungen
gui.custom_monitor_profile	(not set)	Zei-chenket-te	Pfad zum Benutzer-Bildschirmfarbprofil
gui.soft_proofing_profile	(not set)	Zei-chenket-te	Pfad zu Soft-Proof ICC-Farbprofil
gui.icc_custom_monitor_active	false	Boolean	Benutzer-Bildschirmfarbprofil aktiv
gui.icc_soft_proofing_active	false	Boolean	Ausgabe Proof-Farbprofil aktiv
gui.custom_RGB_ICC_profile	(not set)	Zei-chenket-te	Pfad zum Benutzer RGB-ICC-Farbprofil
gui.custom_gray_ICC_profile	(not set)	Zei-chenket-te	Pfad zum Benutzer Grau-ICC-Farbprofil
gui.rendering_intent	1	Ganz-zahl	Farbmanagement Renderabsicht
gui.proofing_intent	1	Ganz-zahl	Farbmanagement Soft-Proof-Renderabsicht
gui.export_intent	1	Ganz-zahl	color management export intent
gui.default_to_srgb	true	Boolean	Nutze sRGB als Default bei nicht farbverwalteten Bildern
gui.working_gamut	0	Ganz-zahl	color management working gamut
gui.export_8bit_method	0	Ganz-zahl	color management export profile for 8bit files
gui.export_16bit_method	1	Ganz-zahl	color management export profile for 16bit files
gui.icc_autoconversion	0	Ganz-zahl	Bilder mit ICC-Farbprofilen automatisch in den Arbeitsfarbraum konvertieren
gui.icc_autoassignment	4	Ganz-zahl	Kodiert Optionen für die automatische Zuweisung von ICC-Profilen
gui.icc_rendering_bpc	true	Boolean	Render-BPC aktivieren
gui.icc_pedantic_linear	false	Boolean	Lineare ICC-Farbprofile pedantisch zuweisen
gui.mouse_actions		Liste von Zei-chenket-ten	Mausaktionen-Konfiguration. Unklare Daten-struktur, bearbeiten Sie diese über die GUI

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Variable	Standard-wert ([Werte-bereich])	Typ	Kommentar
gui.scroll_actions		Liste von Zei-chenket-ten	Scroll-Aktionen-Konfiguration. Unklare Daten-struktur, bearbeiten Sie diese über die GUI
gui_astrometry.compass_position	1 [0, 5]	Ganz-zahl	Index der Kompassposition über dem Gitter
gui_astrometry.cat_messier	true	Boolean	Zeige Messierobjekte in Bemerkungen
gui_astrometry.cat_ngc	true	Boolean	NGC-Objekte in Beschriftungen anzeigen
gui_astrometry.cat_ic	true	Boolean	IC-Objekte in Beschriftungen anzeigen
gui_astrometry.cat_ldn	true	Boolean	LDN-Objekte in Beschriftungen anzeigen
gui_astrometry.cat_sh2	true	Boolean	SH2-Objekte in Beschriftungen anzeigen
gui_astrometry.cat_stars	true	Boolean	Sterne in Beschriftungen anzeigen
gui_astrometry.cat_const	true	Boolean	Zeige Sternbilder in Beschriftungen
gui_astrometry.cat_const_names	true	Boolean	Zeige Sternbildnamen in Beschriftungen
gui_astrometry.cat_user_dso	true	Boolean	Zeige Benutzer-DeepSky-Objekte in Beschriftun-gen
gui_astrometry.cat_user_sso	true	Boolean	Zeige Benutzerobjekte des Sonnensystems in Be-schriftungen
gui_pixelmath.pm_presets		Liste von Zei-chenket-ten	Liste der Pixelmath Vorgaben
script_editor.highlight_syntax	true	Boolean	Hervorheben der Syntax im Skripteditor
script_editor.highlight_bracketma	true	Boolean	Markieren der passende Klammern im Skripted-itor
script_editor.rmargin	true	Boolean	Rechten Rand im Skripteditor anzeigen
script_editor.rmargin_pos	80	Ganz-zahl	Position des rechten Rands im Skripteditor
script_editor.show_linenums	true	Boolean	Zeilennummern im Skripteditor anzeigen
script_editor.show_linemarks	false	Boolean	Zeilenmarkierungen im Skripteditor anzeigen
script_editor.highlight_currentline	true	Boolean	Die aktuelle Zeile im Skripteditor hervorheben
script_editor.autoindent	true	Boolean	Zeilen automatisch einrücken
script_editor.indentontab	true	Boolean	Ausgewählte Blöcke von Zeilen im Scripteditor mit der Tab-Taste einrücken
script_editor.smartbs	true	Boolean	Smart-Backspace-Verhalten im Scripteditor
script_editor.smarthomeend	true	Boolean	Smartes Home / Ende-Verhalten im Scripteditor
script_editor.showspaces	false	Boolean	Sichtbare Leerzeichen und Tabulatorzeichen im Scripteditor anzeigen
script_editor.shownewlines	false	Boolean	Sichtbare Zeilenumbruchzeichen im Scripteditor anzeigen
script_editor.minimap	false	Boolean	Eine Mini-Karte im Scripteditor anzeigen

(*). Für kstars-Kataloge wird dies standardmäßig `~/.local/share/kstars/` sein, unabhängig von Ihrem Betriebssystem.

In jedem Fall müssen Sie sie herunterladen und den von Ihnen gewählten Pfad festlegen.

Siehe Abschnitt über *Verwendung lokaler Kataloge*.

Die Werte können mit dem Befehl `get` abgefragt werden:

Siril Kommandozeile

```
get { -a | -A | variable }
```

Holt einen Wert aus den Einstellungen anhand seines Namens oder listet alle mit **-a** (Liste mit Namen und Werten) oder mit **-A** (detaillierte Liste) auf

Siehe auch SET um Werte zu ändern

Verweise: *set*

Die Werte können mit dem Befehl `set` geändert werden:

Siril Kommandozeile

```
set { -import=inifilepath | variable=value }
```

Aktualisieren eines Einstellungswerts unter Verwendung seines Variablennamens mit dem angegebenen Wert oder einer Reihe von Werten unter Verwendung einer vorhandenen ini-Datei mit der Option **-import=**.
Siehe GET, um Werte oder die Liste der Variablen zu erhalten

Verweise: *get*

KAPITEL 5

Dateiformate

Siril kann auf unterschiedliche Dateiformate öffnen und verarbeiten. Allerdings werden nur zwei Formate nativ gelesen und erlauben die Erstellung von *Sequenzen*: Das FITS und das SER-Format.

Hier werden wir uns die verschiedenen Dateiformate anschauen, die von Siril gelesen werden, und die Limitierungen einiger Dateiformate und die Stärken anderer Formate verstehen.

5.1 Bittiefe

Die Bittiefe gibt die Anzahl der Bits an, die zur Angabe der Farbe eines einzelnen Pixels verwendet werden, oder die Anzahl der Bits, die für jede Farbkomponente eines einzelnen Pixels verwendet werden.

Für Bilder des täglichen Lebens sind 8 Bit mehr als ausreichend. Das bedeutet, dass ein Pixel mit Werten im Bereich [0, 255] kodiert wird. Das Fotografieren von astronomischen Objekten ist jedoch anspruchsvoller und erfordert in der Regel die Bearbeitung von Bildern mit einer Bittiefe von mindestens 16 Bit: *d.h.* im Bereich [0, 65535]. Noch besser ist eine 32-Bit-Präzision, mit der auch die feinsten Informationen erhalten bleiben. Bei diesem letzten Typ werden die Pixel entweder im Intervall [0, 4294967295] oder, wie in Siril verwendet, zwischen den Gleitkommawerten [0, 1] kodiert. Es ist möglich, Formate zu finden, die Pixel auf 64-Bit (im Bereich [0, 1]) kodieren, aber sie sind selten und haben eine sehr spezifische Verwendung. Insbesondere das FITS-Format ermöglicht dies.

Allerdings unterstützen nicht alle Bilddateiformate 16-Bit, geschweige denn 32-Bit. Dies muss daher bei der Wahl des Formats, mit dem Sie arbeiten wollen, berücksichtigt werden.

5.2 Gebräuchliche Dateiformate

Die hier vorgestellten Bilddateiformate sind Standardformate, die von allen Bildbearbeitungsprogrammen gelesen werden können. Diese Formate wurden vor einiger Zeit entwickelt, um den Anforderungen gerecht zu werden, und können veraltet sein. Außerdem wurde keines dieser Formate für die Verarbeitung astronomischer Daten entwickelt. Sie müssen daher im Allgemeinen am Ende der Verarbeitungskette verwendet werden.

Abb. 1: Lineares Bild, gespeichert in 16 Bit

Abb. 2: Dasselbe lineare Bild wurde in 8-Bit gespeichert. Fast alle Daten sind verloren gegangen

5.2.1 AVIF Format

AVIF ist ein modernes Bildformat, das auf dem AV1-Videocodec basiert. Dieses Format verwendet das HEIC-Containerformat. Siril unterstützt das Lesen von AVIF-Bildern mit einer Bittiefe von bis zu 12 Bit. Abhängig von den im installierten libheif kompilierten Codecs ist möglicherweise verlustfreie Unterstützung verfügbar. AVIF unterstützt Farbprofile, die sowohl eingebettete ICC-Profile als auch NCLX-Kennungen verwenden: Eingebettete ICC-Profile werden vollständig unterstützt, und es gibt teilweise Unterstützung für NCLX-Kennungen (sie werden in das entsprechende ICC-Profil konvertiert, obwohl derzeit nur die häufigsten NCLX-Profile unterstützt werden).

Die Dateinamenerweiterung für dieses Format ist **.avif**.

5.2.2 BMP Format

Dateien mit der Erweiterung **.bmp** sind Bitmap-Bilddateien, die zum Speichern digitaler Bitmap-Bilder verwendet werden. Diese Bilder sind unabhängig von der Grafikkarte und werden auch als Device Independent Bitmap (DIB) Dateiformat bezeichnet. Dank dieser Unabhängigkeit kann die Datei auf verschiedenen Plattformen wie Microsoft Windows und Mac geöffnet werden. Das BMP-Dateiformat ermöglicht die Speicherung von Daten als zweidimensionale digitale Bilder, sowohl in Schwarzweiß als auch in Farbe, mit unterschiedlichen Farbtiefen.

Heutzutage wird dieses Format nicht mehr wirklich verwendet und andere Dateitypen werden bevorzugt.

5.2.3 HEIF Format

HEIF ist ein modernes Bildformat, das auf dem x265-Videocodec basiert. Dieses Format verwendet das HEIC-Containerformat. Siril unterstützt das Lesen von HEIC-Bildern mit einer Bittiefe von bis zu 12 Bit. Abhängig von den im installierten libheif kompilierten Codecs ist möglicherweise verlustfreie Unterstützung verfügbar. HEIC unterstützt Farbprofile, die sowohl eingebettete ICC-Profile als auch NCLX-Kennungen verwenden: Eingebettete ICC-Profile werden vollständig unterstützt, und es gibt teilweise Unterstützung für NCLX-Kennungen (sie werden in das entsprechende ICC-Profil konvertiert, obwohl derzeit nur die häufigsten NCLX-Profile unterstützt werden).

Die Dateinamenerweiterung für dieses Format ist **.heic** oder **.heif**.

5.2.4 JPEG Format

Wahrscheinlich das am häufigsten verwendete Dateiformat für den Austausch von Bildern in Foren, per E-Mail oder auf USB-Sticks. Dieses Format ermöglicht eine mehr oder weniger starke (destruktive) Komprimierung, die ideale Dateigrößen für den Datenaustausch ergibt. Die Erweiterung dieses Dateityps ist **.jpg** oder **.jpeg**.

Das JPEG-Format ist jedoch nur in 8 Bit kodiert. Aufgrund der Komprimierung, die Artefakte erzeugt, ist dieses Format für astronomische Bilder nicht sonderlich gut geeignet, und wir bevorzugen im Allgemeinen das PNG-Format.

5.2.5 JPEG XL Format

Nach mehreren früheren Aktualisierungen des JPEG-Formats (JPEG 2000, JPEG XR...) stellt JPEG XL einen großen Leistungssprung dar, mit der erklärten Absicht, alle Rasterformate zu ersetzen. Unabhängig davon, ob dies gelingt oder nicht, verfügt es über mehrere Funktionen, die es für die Astrofotografie interessant machen, und Siril hat daher Lese- und Schreibunterstützung hinzugefügt. Es unterstützt hohe Bittiefen bis zu 32 Bit Fließkomma. Es unterstützt verlustbehaftete und verlustfreie Komprimierung: Bei verlustbehafteter Komprimierung erreicht es bei gleicher Qualitätseinstellung deutlich bessere Komprimierungsraten als JPEG und bei verlustfreier Komprimierung deutlich bessere Komprimierungsraten als PNG. Beachten Sie, dass FITS-Metadaten beim Speichern als JPEG XL nicht erhalten bleiben: Ihre primäre Verwendung in Siril ist wahrscheinlich die Verwendung als Archivformat für endgültige Bearbeitungen von Bildern. Es kann auch für den Web-Export nützlich sein: Das Format wird von Browsern noch nicht umfassend unterstützt, aber Safari unterstützt es nativ und die Unterstützung ist experimentell oder über Add-ons in Firefox und Chrome verfügbar, sodass die Unterstützung in Zukunft möglicherweise breiter verfügbar sein wird .

JPEG XL bietet volle Unterstützung für eingebettete ICC-Profile. Um die Komprimierung zu verbessern, speichert es jedoch erkannte Profile mit einer codierten Kennung anstelle des vollständigen Profils. Das bedeutet, dass die ICC-

Profilbeschreibung, die Sie nach dem Öffnen einer gespeicherten JPEG XL-Datei sehen, möglicherweise nicht mit der ursprünglichen Beschreibung übereinstimmt, das ICC-Profil jedoch funktional identisch ist.

Die Dateinamenerweiterung für dieses Format ist **.jxl**.

5.2.6 PNG Format

Portable Network Graphics ist ein Dateiformat für Rastergrafiken, das eine verlustfreie Datenkompression unterstützt. Die Erweiterung des Formats ist **.png**. PNG-Graustufenbilder unterstützen die größte Bandbreite an Pixeltiefen aller Bildtypen. Es werden Tiefen von 1, 2, 4, 8 und 16 Bit unterstützt, die von einfachen Schwarz-Weiß-Scans bis hin zu medizinischen und astronomischen Bildern mit voller Tiefe reichen.

Kalibrierte astronomische Bilddaten werden in der Regel als 32-Bit- oder 64-Bit-Gleitkommawerte gespeichert, und einige Rohdaten werden als 32-Bit-Ganzzahlen dargestellt. Beide Formate werden von PNG nicht direkt unterstützt.

Dieses Format eignet sich jedoch hervorragend zum Speichern des endgültigen Bildes nach der Bearbeitung.

5.2.7 TIFF Format

TIFF oder TIF, Tagged Image File Format, steht für Rasterbilder zur Verwendung auf verschiedenen Geräten, die diesem Dateiformatstandard entsprechen. Es ist in der Lage, Schwarz-Weiß-, Graustufen-, Palettenfarben- und Vollfarbbilddaten in mehreren Farbräumen zu beschreiben. Es unterstützt sowohl verlustfreie als auch verlustbehaftete Komprimierungsverfahren, damit Anwendungen, die dieses Format verwenden, zwischen Platzbedarf und Verarbeitsgeschwindigkeit wählen können. Die Erweiterung ist entweder **.tiff** oder **.tif**.

Das TIFF-Format hat viele Vorteile. Es unterstützt eine Kodierung von bis zu 32 Bit pro Pixel und bietet eine Vielzahl von möglichen Feldern in den Metadaten, was es zu einem guten Kandidaten für die Speicherung astronomischer Daten macht.

Unter Verwendung des TIFF-Formats und in Zusammenarbeit mit anderen Entwicklern haben wir einen Pseudo-Standard geschaffen, *Astro-TIFF*.

5.2.8 NetPBM Format

Im Rahmen des Netpbm-Projekts werden mehrere Grafikformate verwendet und definiert. Das Portable Pixmap Format (PPM), das Portable Graymap Format (PGM) und das Portable Bitmap Format (PBM) sind Bilddateiformate, die für den einfachen Austausch zwischen verschiedenen Plattformen konzipiert sind. Mögliche Dateierweiterungen sind **.pbm**, **.pgm** (für Schwarz-Weiß- und Graustufendateien) und **.ppm**.

Diese Formate, die bis zu 16 Bit pro Kanal unterstützen, werden kaum genutzt und sollten nur für die endgültige Bildspeicherung verwendet werden.

5.2.9 AVI format

Es ist ein Filmcontainer, der Daten mit verschiedenen Audio- und Videocodecs enthalten kann. Es gibt einige verlustfreie Videocodecs, die in der Vergangenheit für astronomische Aufnahmen verwendet wurden, aber es ist ein Format, das keine für die Astronomie nutzbaren Metadaten enthält, das auf 8-Bit-Bilder beschränkt ist und das keine Garantie dafür bietet, dass die darin enthaltenen Daten unverfälscht sind.

Warnung

Dieses Eingabedateiformat ist jetzt veraltet. Wir empfehlen, stattdessen das Format *SER* zu verwenden.

5.3 FITS Format

5.3.1 Spezifikation

FITS steht für **Flexible Image Transport System** und ist das Standardformat für astronomische Daten, das von professionellen Wissenschaftlern wie der NASA verwendet wird. FITS ist viel mehr als ein Bildformat (wie JPG oder TIFF) und wurde in erster Linie für die Speicherung wissenschaftlicher Daten entwickelt, die aus mehrdimensionalen Arrays bestehen.

Eine FITS-Datei besteht aus einer oder mehreren Header- und Dateneinheiten (HDUs), wobei die erste HDU als "primäre HDU" oder "primäres Array" bezeichnet wird. Es werden fünf primäre Datentypen unterstützt: 8-Bit-Bytes ohne Vorzeichen, 16- und 32-Bit-Ganzzahlen mit Vorzeichen sowie 32- und 64-Bit-Gleitkommazahlen mit einfacher und doppelter Genauigkeit. Das FITS-Format kann auch 16- und 32-Bit-Ganzzahlen ohne Vorzeichen speichern.

Jeder Headerblock besteht aus einer beliebigen Anzahl von 80-stelligen Schlüsselwortsätzen, die die folgende allgemeine Form haben:

```
KEYNAME = value / comment string
```

Der Name des Schlüsselworts kann bis zu 8 Zeichen lang sein und darf nur Großbuchstaben, die Ziffern 0-9, den Bindestrich und den Unterstrich enthalten. Auf den Schlüsselwortnamen folgen (in der Regel) ein Gleichheitszeichen und ein Leerzeichen (=) in den Spalten 9 - 10 des Datensatzes, gefolgt von dem Wert des Schlüsselworts, der entweder eine Ganzzahl, eine Gleitkommazahl, eine Zeichenkette (in einfachen Anführungszeichen) oder ein boolescher Wert (der Buchstabe **T** oder **F**) sein kann.

Das letzte Schlüsselwort im Header ist immer das Schlüsselwort `END`, das keine Wert- oder Kommentarfelder enthält.

Jeder Headerblock beginnt mit einer Reihe von erforderlichen Schlüsselwörtern, die die Größe und das Format der folgenden Dateneinheit angeben. Der Header eines 2-dimensionalen primären Bildarrays beginnt beispielsweise mit den folgenden Schlüsselwörtern:

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                   16 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                  440 / length of data axis 1
NAXIS2  =                  300 / length of data axis 2
```

Bemerkung

In Siril werden 64-Bit-FITS-Dateien nicht unterstützt. Siril liest sie, konvertiert sie aber in 32-Bit-Dateien.

5.3.2 Filenames

FITS files should have an extension ".fit", ".fits" or ".fts"; uppercase variations are accepted but not recommended. It is possible to select your preferred extension in the *Preferences* dialog, and this will be used for saving FITS files, but all permitted extensions will be accepted for reading FITS images or sequence frames.

Warnung

The FITS file library cfitsio interprets square brackets in filenames as providing filtering options. Therefore regardless of whether square bracket characters are permitted in filenames by your operating system, they **must not** be used in FITS files.

5.3.3 Komprimierung

Die Komprimierung ist eine Methode zur Verringerung der Dateigröße. Es gibt viele Komprimierungsmethoden, je nach Art der verwendeten Bilder. Diese Kompression kann destruktiv sein, wie bei JPEG, oder verlustfrei, wie bei PNG.

Es ist möglich, mit komprimierten FITS-Dateien zu arbeiten. Auf Kosten einer längeren Berechnungszeit kann die Größe der Dateien erheblich reduziert werden. Siril bietet mehrere Komprimierungsalgorithmen an, die im Folgenden aufgeführt sind:

- **Rice**: Der Rice-Algorithmus ist einfach und sehr schnell
- **GZIP 1**: Der gzip-Algorithmus wird zum Komprimieren und Dekomprimieren der Bildpixel verwendet. Gzip ist der Kompressionsalgorithmus, der in dem gleichnamigen freien GNU-Softwareprogramm verwendet wird.
- **GZIP 2**: Die Bytes im Array der Bildpixelwerte werden in abnehmender Reihenfolge ihrer Bedeutung gemischt, bevor sie mit dem gzip-Algorithmus komprimiert werden. Dies ist normalerweise besonders effektiv bei der Komprimierung von Fließkomma-Arrays.

Diesen Algorithmen ist eine Option zugeordnet, die **Quantisierungsstufe**:

Während Bilder im Fließkommaformat verlustfrei komprimiert werden können (mit gzip, da Rice nur ganzzahlige Arrays komprimiert), lassen sich diese Bilder oft nicht sehr gut komprimieren, weil die Pixelwerte zu verrauscht sind; die weniger signifikanten Bits in der Mantisse der Pixelwerte enthalten effektiv nicht komprimierbare Zufalls-Bitmuster. Um eine höhere Komprimierung zu erreichen, muss man einen Teil dieses Rauschens entfernen, ohne jedoch den nützlichen Informationsgehalt zu verlieren. Ist er zu groß, werden die Pixelwerte unterabgetastet (undersampled), was zu einem Informationsverlust im Bild führt. Ist er jedoch zu klein, bleibt zu viel Rauschen in den Pixelwerten erhalten (oder das Rauschen wird sogar verstärkt), was zu einer schlechten Kompimierungsrate führt.

Bemerkung

Die unterstützten Bildkomprimierungsalgorithmen sind alle **verlustfrei**, wenn sie auf ganzzahlige FITS-Bilder angewandt werden; die Pixelwerte bleiben während des Komprimierungs- und Dekomprimierungsprozesses exakt und ohne Informationsverlust erhalten. Fließkomma-FITS-Bilder (mit `BITPIX = -32` oder `-64`) werden zunächst in skalierte Integer-Pixelwerte quantisiert, bevor sie komprimiert werden. Diese Technik führt zu viel höheren Kompressionsfaktoren als die einfache Verwendung von GZIP zur Komprimierung des Bildes, bedeutet aber auch, dass die ursprünglichen Fließkomma-Pixelwerte möglicherweise nicht genau wiedergegeben werden, wenn das Bild entpackt wird. Wenn dies richtig gemacht wird, wird nur das "Rauschen" der Fließkommawerte entfernt, ohne dass wichtige Informationen verloren gehen.

5.3.4 Particular Cases of Scientific FITS Files

FITS files produced by major astronomical missions such as those from NASA's Hubble Space Telescope (HST) or the James Webb Space Telescope (JWST) are often more complex than typical FITS files used by amateur astronomers. These files are generally structured as FITS sequences (also referred to as FITS cubes), but with an important difference: the individual images within these sequences may have different dimensions.

This feature allows for storing multi-dimensional data collected under varying conditions, where each frame or image in the sequence might not have the same resolution or size. For example, data from the JWST often involves such heterogeneous datasets due to the nature of its instruments and observing modes.

In Siril, when processing these types of scientific FITS files, it is crucial to enable the option that allows handling images of varying dimensions within a FITS cube. This can be done by selecting the option *Allow FITS cubes to have images of different size* in the *FITS Options* tab of the preferences.

> **Warnung**
>
> Currently, Siril is unable to process FITS files with `NAXES[2] > 3`. This feature is planned for future versions of the software, so FITS cubes with more than three axes are not supported at the moment.

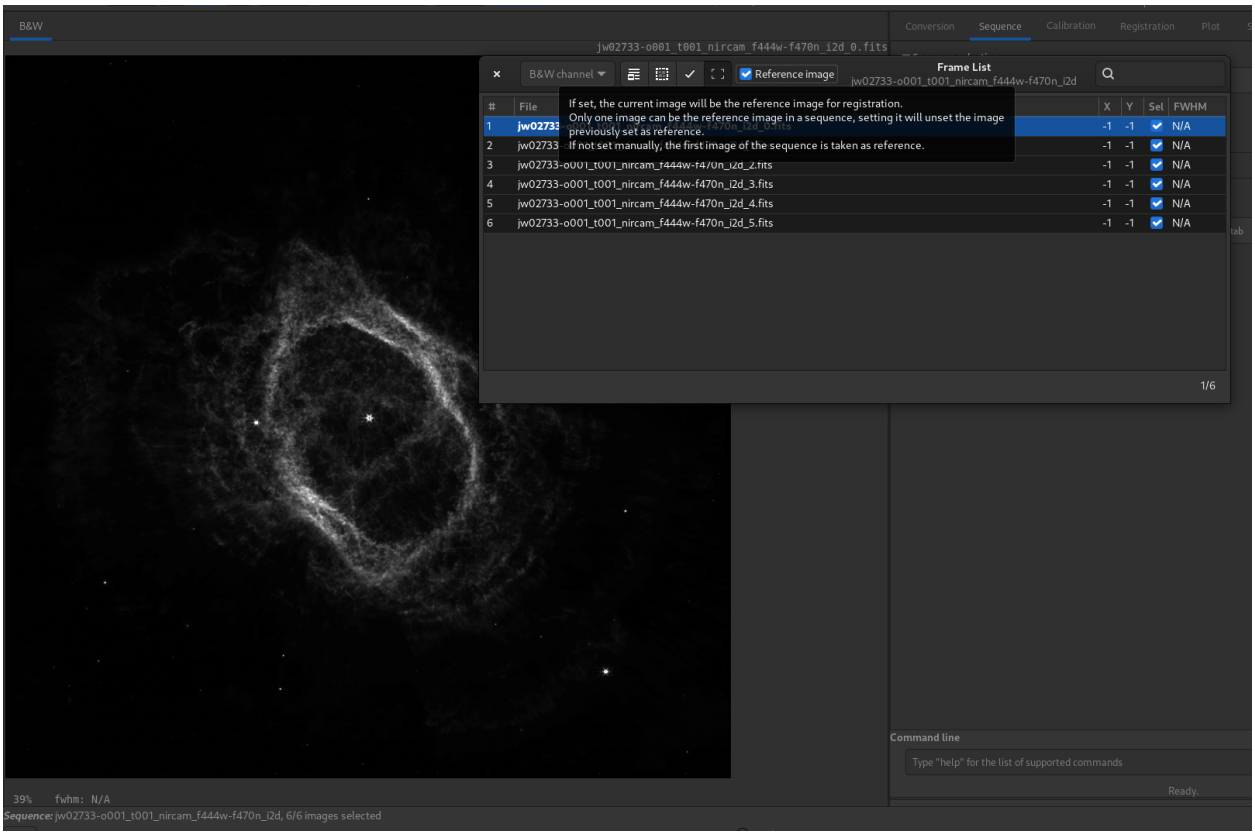


Abb. 3: Beispiel für ein in Siril geladenes JWST-Bild

5.3.5 Ausrichtung von FITS-Bildern

Der FITS-Standard ist ein Container, der beschreibt, wie Bilddaten und Metadaten gespeichert werden können. Professionelle Werkzeuge aus der Anfangszeit des FITS-Formats, wie ds9 (Harvard Smithsonian Center for Astrophysics), fv (FITS-Viewer der NASA), speichern Bilder **von unten nach oben**. Man könnte versucht sein zu sagen, dass dies keine Rolle spielt, aber wenn es um Debayern oder Astrometrie geht, treten Probleme auf. Zum Beispiel wird das übliche **RGGB** Bayer-Muster zu **GBRG**, wenn das Bild auf dem Kopf steht.

Heutzutage schreiben die meisten Kameratreiber die Daten trotzdem in der Reihenfolge von oben nach unten, und wir müssen damit zurechtkommen.

Aus diesen Gründen haben wir kürzlich zusammen mit P. Chevalley von CCDCiel ein **neues FITS-Schlüsselwort** eingeführt. Wir ermutigen alle Datenproduzenten, INDI- und ASCOM-Entwickler, es zu verwenden, um die Dinge für alle einfacher zu machen.

Dieses Schlüsselwort ist `ROWORDER` vom Typ `TSTRING`. Es kann zwei Werte annehmen: `BOTTOM-UP` und `TOP-DOWN`.

Siril liest und zeigt Bilder immer in der Reihenfolge von unten nach oben an. Wenn jedoch im Schlüsselwort Top-Down-Informationen angegeben sind, wird Siril das Bild mit dem korrigierten Muster debayert.

Warum schreiben manche Programme Bilder überhaupt von unten nach oben?

Der Grund dafür ist: Mathematik macht es so.

Außerdem sagt die FITS-Spezifikation:

> ***5.1. Konventionen für die Bildanzeige***
>
> *Es ist sehr hilfreich, eine Konvention für die Darstellung von Bildern, die im FITS-Format übertragen werden, festzulegen. Viele der aktuellen Bildverarbeitungssysteme haben sich auf eine solche Konvention geeinigt. Daher empfehlen wir den FITS-Schreibern, die Pixel so anzuordnen, dass das erste Pixel in der FITS-Datei (für jede Bildebene) dasjenige ist, das vom Bildverarbeitungssystem des FITS-Schreibers in der linken unteren Ecke angezeigt würde (wobei die erste Achse nach rechts und die zweite Achse nach oben ansteigt). Diese Konvention ist in Ermangelung einer Beschreibung der Weltkoordinaten eindeutig hilfreich. Sie schließt nicht aus, dass ein Programm sich die Achsenbeschreibungen ansieht und diese Konvention außer Kraft setzt, oder dass der Benutzer eine andere Darstellung anfordert. Diese Konvention entbindet die FITS-Autoren auch nicht davon, vollständige und korrekte Beschreibungen der Bildkoordinaten zu liefern, die es dem Benutzer ermöglichen, die Bedeutung des Bildes zu bestimmen. Die Anordnung des Bildes für die Anzeige ist lediglich eine Konvention der Bequemlichkeit, während die Koordinaten der Pixel Teil der Physik der Beobachtung sind.*

> **Warnung**
>
> Das Schlüsselwort `ROWORDER` kann verwendet werden für:
>
> 1. Anzeige des Bildes in der gewünschten Ausrichtung (Spiegeln der Anzeige).
> 2. Spiegeln des Bayer-Demosaikmusters. So kann das Demosaikmuster entsprechend dem Sensorhersteller festgelegt werden.
>
> **ABER**
>
> 1. Der Befehl `ROWORDER` darf nicht verwendet werden, um die Bilddaten für das Stacking zu spiegeln. Andernfalls würden neue Bilder mit älteren Dark- und Flatbildern inkompatibel werden.
> 2. „ROWORDER“ darf nicht verwendet werden, um die Bilddaten für die astrometrische Lösung zu spiegeln. Dies würde dazu führen, dass die astrometrische Lösung nicht mit anderen Programmen kompatibel ist.

5.3.6 Software, die das Schlüsselwort ` ROWORDER` verwendet

- Siril (ab Version 0.99.4)
- CCDCiel (seit Version 0.9.72)
- Indi (seit Juli 2020)
- KStars (seit 3.4.3)
- SharpCap (seit Version 3.3)
- FireCapture (seit Version 2.7)
- N.I.N.A (seit Version 1.10)
- MaxImDL (seit Version 6.23)
- INDIGO (seit Jul. 2020)

- PixInsight (seit Version 1.8.8-6)
- ASTAP (seit Version ß0.9.391)
- APT (seit Version 3.86.3)
- AstroDMx Capture (seit Version 0.80)
- Astroart (seit Version 8.0)

5.3.7 Abrufen der Bayer-Matrix

Die Reihenfolge der Bildzeilen ändert die Art und Weise, wie die Bayer-Matrix gelesen werden sollte, aber es gibt auch zwei optionale FITS-Header-Schlüsselwörter, die sich darauf auswirken: `XBAYROFF` und `YBAYROFF`. Sie geben einen Offset zur Bayer-Matrix an, um das Lesen in der ersten Spalte oder der ersten Zeile zu beginnen.

Um Entwicklern bei der Integration der Schlüsselwörter „ROWORDER“, „XBAYROFF“ und „YBAYROFF“ in ihre Software zu helfen, wurden einige Testbilder von Han Kleijn von „hnsky.org“ <https://www.hnsky.org > erstellt`_, eines für jede Kombination der drei Schlüsselwörter. Laden Sie sie hier herunter: Bayer_test_pattern_v6.tar.gz.

5.3.8 Liste der FITS-Schlüsselwörter

Siril kann eine Vielzahl von Schlüsselwörtern lesen und interpretieren. Die folgende Liste veranschaulicht die nicht standardmäßigen Schlüsselwörter, die Siril bei Bedarf erkennt. Einige von Siril gelesene Schlüsselwörter erscheinen möglicherweise nicht in dieser Liste. Beispielsweise werden die Schlüsselwörter „CCDTEMP“ oder „TEMPERAT“, die die Temperatur des Sensors angeben, korrekt gelesen, aber unter dem Schlüsselwort „CCD-TEMP“ weitergegeben.

Tipp

Siril kann Prüfsummen lesen und vergleichen, wenn sie im FITS-Header vorhanden sind. Standardmäßig und aus Gründen der Softwareleistung werden jedoch `CHECKSUM`- und `DATASUM`-Karten automatisch aus den HDU-Headern entfernt, wenn eine Datei geöffnet wird, und alle `CHECKSUM`- oder `DATASUM`-Karten werden aus den Headern entfernt, wenn eine HDU in eine Datei geschrieben wird. Sie können jedoch auf Anforderung des Benutzers mit der Option **-chksum** des Befehls *save* oder über die grafische Benutzeroberfläche gespeichert werden.

Tab. 1: Von Siril gespeicherte FITS-Schlüsselwörter. Aus Gründen der Übersichtlichkeit werden die SIP-Schlüsselwörter für die Astrometrie nicht aufgeführt.

FITS Schlüsselwort	Typ	Kommentar
BZERO	Double	Datenbereich Offset zu dem von Vorzeichenlosen Shorts verschieben
BSCALE	Double	Standard-Skalierungsfaktor
MIPS-HI	Unsigned short	Oberer Grenzwert der Visualisierung
MIPS-LO	Unsigned short	Unterer Grenzwert der Visualisierung
MIPS-FHI	Float	Oberer Grenzwert der Visualisierung
MIPS-FLO	Float	Unterer Grenzwert der Visualisierung
PROGRAM	String	Software, die diese HDU erstellt hat
FILENAME	String	Originaler Dateiname
DATE	String	Datum zu dem das FITS erstellt wurde als UTC-Datum
DATE-OBS	String	YYYY-MM-DDThh:mm:ss Beobachtungsbeginn, UT
IMAGETYP	String	Bildtyp
ROWORDER	String	Reihenfolge der Zeilen in einem Bildarray
EXPTIME	Double	[s] Belichtungszeit

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

FITS Schlüsselwort	Typ	Kommentar
TELESCOP	String	Name des für die Aufnahme genutzten Teleskops
OBSERVER	String	Name des Beobachters
FILTER	String	Name des genutzten Filters
APERTURE	Double	Blende des Instruments
ISOSPEED	Double	ISO-Einstellung der Kamera
FOCALLEN	Double	[mm] Brennweite
CENTALT	Double	[deg] Höhe/Alt des Teleskops
CENTAZ	Double	[deg] Azimuth des Teleskops
XBINNING	Unsigned int	Binning-Modus der Kamera
YBINNING	Unsigned int	Binning-Modus der Kamera
XPIXSZ	Double	[um] Pixelgröße der X-Achse
YPIXSZ	Double	[um] Pixelgröße der Y-Achse
INSTRUME	String	Name des Instruments
CCD-TEMP	Double	[degC] Temperatur des CCSs
SET-TEMP	Double	[degC] Eingestellte Temperatur des CCDs
GAIN	Unsigned short	Gain der Kamera
OFFSET	Unsigned short	Offset-Einstellung der Kamera
CVF	Double	[e-/ADU] Elektronen pro A/D-Einheit (ADU)
BAYERPAT	String	Bayermuster
XBAYROFF	Int	X-Offset der Bayer-Matrix.
YBAYROFF	Int	Y-Offset der Bayer-Matrix
FOCNAME	String	Name des Fokusers
FOCPOS	Int	[step] Position des Fokusers
FOCUSSZ	Int	[um] Schrittweite des Fokusers
FOCTEMP	Double	[degC] Temperatur des Fokusers
STACKCNT	Unsigned int	Anzahl gestackter Einzelbilder
LIVETIME	Double	[s] Belichtungszeit nach Totzeitkorrektur
EXPSTART	Double	[JD] Startzeitpunkt der Belichtung (standardisiertes julianisches Datum)
EXPEND	Double	[JD] Endzeitpunkt der Belichtung (standardisiertes julianisches Datum)
OBJECT	String	Name des interessierenden Objekts
AIRMASS	Double	Luftmasse (Airmass) in der Bildmitte (Gueymard 1993)
SITELAT	Double	[Grad] Breitengrad des Beobachtungsortes
SITELONG	Double	[Grad] Längengrad des Beobachtungsortes
SITEELEV	Double	[m] Höhe des Beobachtungsortes
DFTTYPE	String	Modul/Phase einer diskreten Fourier-Transformation
DFTORD	String	Niedrige/hohe Raumfrequenz. befinden sich in der Bildmitte
DFTNORM1	Double	Normalisierungswert für Kanal #1
DFTNORM2	Double	Normalisierungswert für Kanal #2
DFTNORM3	Double	Normalisierungswert für Kanal #3
CTYPE3	String	RGB-Bild
OBJCTRA	String	Bildmitte Rektaszension (hms)
OBJCTDEC	String	Bildmitte Deklination (dms)
RA	Double	Bildmitte Rektaszension (Grad)
DEC	Double	Bildmitte Deklination (Grad)
CTYPE1	String	TAN (gnomische) Projektion
CTYPE2	String	TAN (gnomische) Projektion
CUNIT1	String	TAN (gnomische) Projektion und SIP-Verzerrungen

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

FITS Schlüsselwort	Typ	Kommentar
CUNIT2	String	TAN (gnomische) Projektion und SIP-Verzerrungen
EQUINOX	Double	Äquatoriales Äquinoktium
CRPIX1	Double	Referenzpixel für Achse1
CRPIX2	Double	Referenzpixel für Achse2
CRVAL1	Double	Referenzwert für Achse1 (Grad)
CRVAL2	Double	Referenzwert für Achse2 (Grad)
LONPOLE	Double	Nativer Längengrad des Himmelspols
CDELT1	Double	Pixelgröße X (Grad)
CDELT2	Double	Pixelgröße Y (Grad)
PC1_1	Double	Lineare Transformationsmatrix (1, 1)
PC1_2	Double	Lineare Transformationsmatrix (1, 2)
PC2_1	Double	Lineare Transformationsmatrix (2, 1)
PC2_2	Double	Lineare Transformationsmatrix (2, 2)
CD1_1	Double	Skalierungsmatrix (1, 1)
CD1_2	Double	Skalierungsmatrix (1, 2)
CD2_1	Double	Skalierungsmatrix (2, 1)
CD2_2	Double	Skalierungsmatrix (2, 2)
PLTSOLVD	Logical	SIRIL Solver (astrometrischer Löser)

5.4 Astro-TIFF

Im Jahr 2022 bot Han Kleijn, der Entwickler der Software ASTAP, an, an der Entwicklung eines neuen Pseudo-Standards mitzuwirken, der das TIFF-Format verwendet und die Leistungsfähigkeit der FITS-Header nutzt. So entstand das Format Astro-TIFF.

5.4.1 Warum ein neues Dateiformat unter all den Anderen?

Das derzeit am häufigsten verwendete Format für die Astrofotografie ist das FITS-Format. Dieses Format wurde von professionellen Wissenschaftlern entwickelt und erfüllt alle Erwartungen von Amateuren. Und obwohl seine große Flexibilität einige Bedenken hinsichtlich der Kompatibilität hervorruft, bleibt es das bevorzugte Format.

Es gibt noch andere spezialisierte Formate, die jedoch meist einer bestimmten Software zugeordnet sind. Wie das vom PixInsight-Team entwickelte XISF-Format. Dieses letzte Format wird zwar häufig in Siril angefordert, ist jedoch ein Format, das PixInsight, einer proprietären Software, gewidmet ist. Daher ist das Interesse an der Entwicklung der Kompatibilität mit Siril minimal und wir haben dies nur zum Lesen im 1.4.x-Zyklus umgesetzt.

Die Entwicklung von Astro-TIFF erscheint dann als eine gute Alternative, denn basierend auf dem TIFF-Format ist es möglich, die Dateien mit jeder Bildbearbeitungssoftware zu öffnen.

Schließlich unterstützt das TIFF-Format (wie auch das FITS-Format) eine Komprimierung, die eine kleinere Dateigröße ermöglicht.

5.4.2 Spezifikation 1.0

Datiert 2022-06-21

- Die Dateien entsprechen vollständig der TIFF 6.0-Spezifikation einschließlich Zusatz 2.
- Der FITS-Header wird in das TIFF-Basistag **Bildbeschreibung** geschrieben. Code 270, Hex 010E.
- Der Header folgt der FITS-Spezifikation mit der Ausnahme, dass die Zeilen kürzer als 80 Zeichen sein können und die Zeilen entweder mit CR+LF (0D0A) oder LF (0A) enden.

- Die erste Zeile der Beschreibung ist die erste Headerzeile und beginnt mit `SIMPLE`. Die letzte Zeile des Headers beginnt mit `END`.

Empfehlungen

- `TIFFtag_orientation=1` (links oben) Die Ausrichtung folgt den Konventionen. Pixel `FITS_image[1,1]` ist links unten. `TIFF_image[0,0]` ist links oben. Diese Pixel werden zunächst in die Datei geschrieben oder gelesen. Wenn Sie also ein FITS-Bild in TIFF schreiben und dabei die Ausrichtung für den Benutzer beibehalten, ist das erste zu schreibende Pixel `FITS_image[1,NAXIS2]`.
- `TIFFtag_compression=8` (Deflate) oder 5 (LZW).
- Für Graustufenbilder gilt `TIFFtag_PhotometricInterpretation = 1` (Mindestwert ist schwarz, Höchstwert ist weiß).
- Schreiben Sie alle verfügbaren Header-Schlüsselwörter.

Hinweise

- Diese Verwendung des TIFF-Formats ist für 16-Bit-Lights, -Darks, -Flats und -Flat-Darks (astronomische Bilder) gedacht, kann aber auch im 32-Bit-Format verwendet werden. Es ist möglich, FITS in TIFF und umgekehrt zu konvertieren, aber der Anwendungsprogrammierer kann entscheiden, nur im Astro-TIFF-Format zu exportieren (schreiben) oder nur zu importieren (lesen).
- Wenn eine astrometrische Lösung enthalten ist, sollte sie mit der Bildausrichtung übereinstimmen.
- Einige Header-Schlüsselwörter sind redundant wie `NAXIS1`, `NAXIS2`, `BZERO` und `BITPIX` und werden nicht benötigt. TIFF-Bildabmessungen und -typ sind maßgeblich.
- Das in der Kopfzeile angegebene Debayer-Pattern sollte mit der Bildausrichtung übereinstimmen.
- Der Header wird in vielen Bildbearbeitungsprogrammen sichtbar sein.

Beispiel für einen Astro-TIFF-Header, der genauso aussieht wie ein FITS-Header:

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                  -32 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                 6248 / length of data axis 1
NAXIS2  =                 4176 / length of data axis 2
NAXIS3  =                    1 / length of data axis 3
EXTEND  =                    T / FITS dataset may contain extensions
COMMENT   FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT   and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
BZERO   =                    0 / offset data range to that of unsigned short
BSCALE  =                    1 / default scaling factor
DATE    = '2022-12-14T16:05:47' / UTC date that FITS file was created
DATE-OBS= '2022-05-06T20:29:20.019000' / YYYY-MM-DDThh:mm:ss observation start,
INSTRUME= 'ZWO CCD ASI2600MM Pro' / instrument name
OBSERVER= 'Unknown '           / observer name
TELESCOP= 'iOptron ZEQ25'      / telescope used to acquire this image
ROWORDER= 'TOP-DOWN'           / Order of the rows in image array
XPIXSZ  =                 3.76 / X pixel size microns
YPIXSZ  =                 3.76 / Y pixel size microns
XBINNING=                    1 / Camera binning mode
YBINNING=                    1 / Camera binning mode
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
FOCALLEN=              370.092 / Camera focal length
CCD-TEMP=                 -9.8 / CCD temp in C
EXPTIME =                  120 / Exposure time [s]
STACKCNT=                  126 / Stack frames
LIVETIME=                15120 / Exposure time after deadtime correction
FILTER  = 'Lum     '           / Active filter name
IMAGETYP= 'Light Frame'        / Type of image
OBJECT  = 'Unknown '           / Name of the object of interest
GAIN    =                  100 / Camera gain
OFFSET  =                   50 / Camera offset
CTYPE1  = 'RA---TAN'           / Coordinate type for the first axis
CTYPE2  = 'DEC--TAN'           / Coordinate type for the second axis
CUNIT1  = 'deg     '           / Unit of coordinates
CUNIT2  = 'deg     '           / Unit of coordinates
EQUINOX =                 2000 / Equatorial equinox
OBJCTRA = '09 39 54.932'       / Image center Right Ascension (hms)
OBJCTDEC= '+70 00 10.118'      / Image center Declination (dms)
RA      =              144.979 / Image center Right Ascension (deg)
DEC     =              70.0028 / Image center Declination (deg)
CRPIX1  =               3123.5 / Axis1 reference pixel
CRPIX2  =               2088.5 / Axis2 reference pixel
CRVAL1  =              144.979 / Axis1 reference value (deg)
CRVAL2  =              70.0028 / Axis2 reference value (deg)
CD1_1   =         -0.000580606 / Scale matrix (1, 1)
CD1_2   =         -4.12215e-05 / Scale matrix (1, 2)
CD2_1   =         -4.11673e-05 / Scale matrix (2, 1)
CD2_2   =          0.000580681 / Scale matrix (2, 2)
PLTSOLVD=                    T / Siril internal solve
HISTORY Background extraction (Correction: Subtraction)
HISTORY Plate Solve
END
```

5.4.3 Speichern von Astro-TIFF in Siril

In Siril können Sie Astro-TIFF-Dateien speichern, indem Sie das TIFF-Format im Speicherdialog auswählen, wenn Sie auf *Speichern unter* klicken. In der Dropdown-Liste des TIFF-Dialogs können Sie wählen, ob Sie im Standard-TIFF-Format oder im Astro-TIFF-Format speichern möchten. Letzteres ist das Standardformat.

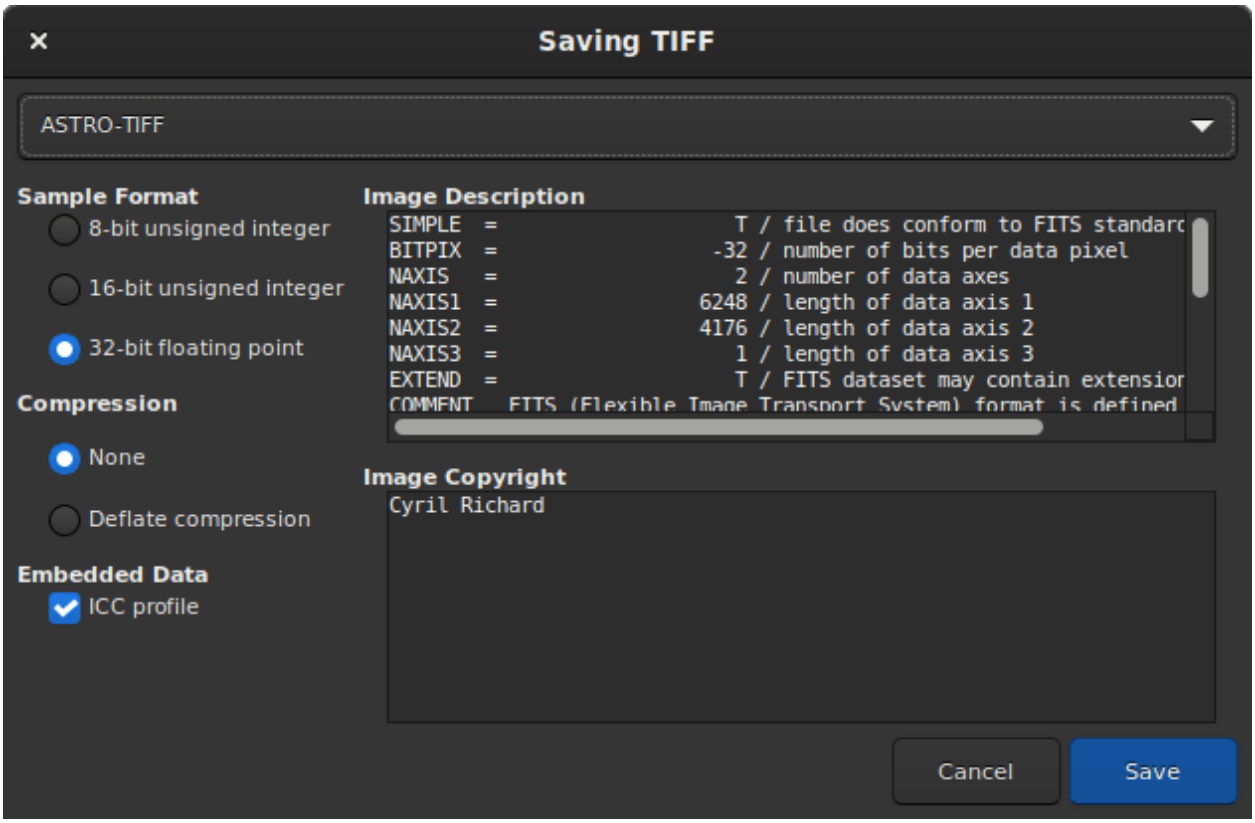


Abb. 4: Speichern-Dialog mit Astro-TIFF-Option

Siril Kommandozeile

```
savetif filename [-astro] [-deflate]
```

Speichert das aktuelle Bild in Form einer unkomprimierten TIFF-Datei mit 16 Bit pro Kanal: **filename**.tif. Die Option **-astro** ermöglicht das Speichern im Astro-Tiff-Format, während **-deflate** die Kompression aktiviert

Siehe auch SAVETIF32 und SAVETIF8

5.4.4 Beispieldateien

- Astro-TIFF-Datei erstellt von Siril (32-bit, unkomprimiert).
- Astro-TIFF-Datei erstellt von Siril (32-bit, komprimiert).

5.5 SER Format

Das SER-Dateiformat ist ein einfaches Bildsequenzformat, ähnlich wie bei unkomprimierten Videos. Die Dokumentation ist auf der offiziellen Seite zu finden. Das neueste PDF-Dokument ist auch auf der gespiegelten freien Seite zu finden.

Mit den Verbesserungen der Versionen 2 und 3 kann SER auch Farbbilder verarbeiten, was es zum perfekten Ersatz für das übliche AVI- oder ein anderes Videoformat macht, das von älteren Aufnahmeprogrammen in allen astronomischen Situationen erzeugt wird. Komprimierte Bilder sollten nicht für die Astronomie verwendet werden, können aber dennoch in SER konvertiert werden, wodurch die Dateien bei gleicher Qualität größer werden, aber einfacher zu bearbeiten sind.

Siril kann jede Bildsequenz und viele Filmformate in SER-Dateien umwandeln. Ser-player ist ein großartiges Tool, mit dem SER-Dateien wie ein Film angezeigt werden können, mit vielen Optionen und funktioniert auf den meisten Betriebssystemen.

Das Hauptproblem bei AVI und anderen Filmcontainern besteht darin, dass sie für viele Codecs und Pixelformate ausgelegt sind, was für allgemeine Filme gut ist, aber von der Astronomiesoftware verlangt, dass sie eine große Anzahl unterschiedlicher Dateiformate verarbeiten kann. Allzweck-Videosoftware ist oft nicht in der Lage, 16-Bit-Pixelwerte oder einige unkomprimierte Datenformate zu verarbeiten. Bei SER reicht ein einziges Dateiformat aus, um alles zu verarbeiten, weshalb Siril zum Beispiel jetzt eine Verarbeitung nur für SER entwickelt.

5.5.1 Dateistruktur

Eine SER-Datei besteht aus drei Teilen:

- einen 178-Byte-Header mit Bild- und Beobachtungsinformationen
- Bilddaten, Rohpixeldaten
- ein optionaler Trailer, der Daten für alle Bilder der Sequenz enthält

5.5.2 Umgang mit Farben

In Version 3 (2014) gibt es zwei Möglichkeiten, farbige Bilder in SER zu verarbeiten. Wenn die Daten direkt von einem Sensor kommen, ist der bevorzugte Weg wahrscheinlich die Verwendung von Ein-Ebenen-Bildern und die Interpolation von Daten aus dem Farbfilter-Matrix (ähnlich den CFA-Dateiformaten, die in Astronomiesoftware verwendet werden).

Die andere Möglichkeit, die in Version 3 hinzugefügt wurde, ist die Verwendung von drei Ebenen zur Darstellung von RGB-Bilddaten. SER v3 unterstützt RGB/BGR 8/16-Bit-Daten. Dies kann nützlich sein, wenn Daten aus einer Quelle mit einem unbekannten Farbfilter-Array konvertiert werden oder für allgemeine Konvertierungen.

5.5.3 Spezifikationsproblem mit Endianness

Da SER-Dateien Bilder mit 16-Bit-Präzision enthalten können, muss die Endianness genau angegeben werden. Die Spezifikation erlaubt es, dass die Endianness entweder Big-Endian oder Little-Endian sein kann, um die Dateierstellung auf verschiedenen Systemen zu erleichtern, solange die verwendete Endianness im Header der Datei dokumentiert ist.

Aus einem unbekannten Grund missachten einige der ersten Programme, die SER unterstützen, die Spezifikation für das Endianness-Flag. Die Spezifikation besagt, dass ein boolescher Wert für den LittleEndian-Header verwendet wird, und sie verwenden ihn als BigEndian-Header, mit 0 für Little-Endian und 1 für Big-Endian. Um die Kompatibilität mit diesen ersten Implementierungen nicht zu verletzen, haben spätere Programme wie Siril, GoQat, Ser-player und viele andere beschlossen, diesen Header entgegen der Spezifikation zu implementieren.

5.6 IRIS PIC

Das PIC-Format ist ein proprietäres Bildformat, das für die IRIS-Software von Christian Buil erstellt wurde. Um die Kompatibilität mit Letzterem sicherzustellen, ist Siril in der Lage, diesen Dateityp zu lesen. Da das Format jedoch proprietär ist und die Spezifikationen nicht bekannt sind, werden bei der Konvertierung in FITS nicht alle Header-Informationen gespeichert.

Siril kann nicht im PIC-Format aufnehmen.

5.7 PixInsight XISF

PixInsight hat eine Dateistruktur namens Extensible Image Serialization Format oder XISF eingeführt. Dieses Format wurde geschrieben, um das in der Astrofotografie häufig verwendete FITS-Format zu ersetzen. Dieses Format wurde jedoch im Wesentlichen von und für die PixInsight-Software entwickelt und „viele wichtige Arten von Metadaten werden in privaten Namespaces gespeichert <https://pixinsight.com/forum/index.php?threads/pcl-astrometricsolution-properties“. .22627/post-144255>`_, die laut den Hinweisen der Pixinsight-Entwickler nicht verlässlich sind sich zwischen den XISF-Spezifikationsversionen willkürlich ändern können. Darüber hinaus kann die Art und Weise, wie ein Schlüsselwort zur Beschreibung einer bestimmten Funktion verwendet wird, „sehr undurchsichtig sein <https://pixinsight.com/forum/index.php?threads/xml-keys.18472/#post-145431>“_. Daher sind die Vorteile, die es für die Verwendung innerhalb von Pixinsight mit sich bringen könnte, für andere Software im Allgemeinen gleich Null. Darüber hinaus ist das FITS-Dateiformat ein anerkanntes und weit verbreitetes Format in der Community. Die NASA selbst nutzt es für ihre eigenen Bilder. Es ist sehr flexibel und leistungsstark und enthält alle Funktionen, die für die moderne astrofotografische Verarbeitung erforderlich sind.

Wir haben uns daher entschieden, Siril das Lesen dieser Art von Dateien zu ermöglichen, allerdings im schreibgeschützten Modus. Eine Speicherung im XISF-Format ist und wird nicht implementiert.

KAPITEL 6

Farbmanagement

Siril bietet jetzt einen Workflow mit vollständigem Farbmanagement mit ICC-Profilen. Da Siril eine genaue und konsistente Farbwiedergabe auf verschiedenen Geräten gewährleistet, können Sie Ihre Bilder in der Gewissheit bearbeiten, dass die Farben, die Sie auf Ihrem Bildschirm sehen, auch auf Ausdrucken oder anderen Displays wiedergegeben werden. Sie können Ihre Bilder in Farbräumen bearbeiten, die besser auf die Fähigkeiten eines professionellen Fotodruckers mit großem Farbraum abgestimmt sind, und so Abzüge mit satteren Farben erstellen, als dies bei der Arbeit im Standardfarbraum sRGB möglich ist. Außerdem können Sie Bilder produzieren, die Ihr Publikum unabhängig von seinem Anzeigegerät genauso sehen kann, wie Sie sie während der Bearbeitung sehen.

Dieser Teil der Dokumentation besteht aus mehreren Seiten, die einen grundlegenden Überblick über die Funktionsweise des Farbmanagements, seine Vorteile und die Funktionsweise in Siril geben sollen.

6.1 Erste Schritte

Willkommen beim neuen Farbmanagementsystem von Siril! Das Farbmanagement kann sehr komplex erscheinen, daher soll diese Seite die absoluten Grundlagen vermitteln, die Sie für den Einstieg benötigen. Um Ihnen das Leben leichter zu machen, werden zwei Einstellungen behandelt.

6.1.1 Standardeinstellung

Die Standardeinstellung bietet ein ähnliches Verhalten wie frühere Versionen von Siril.

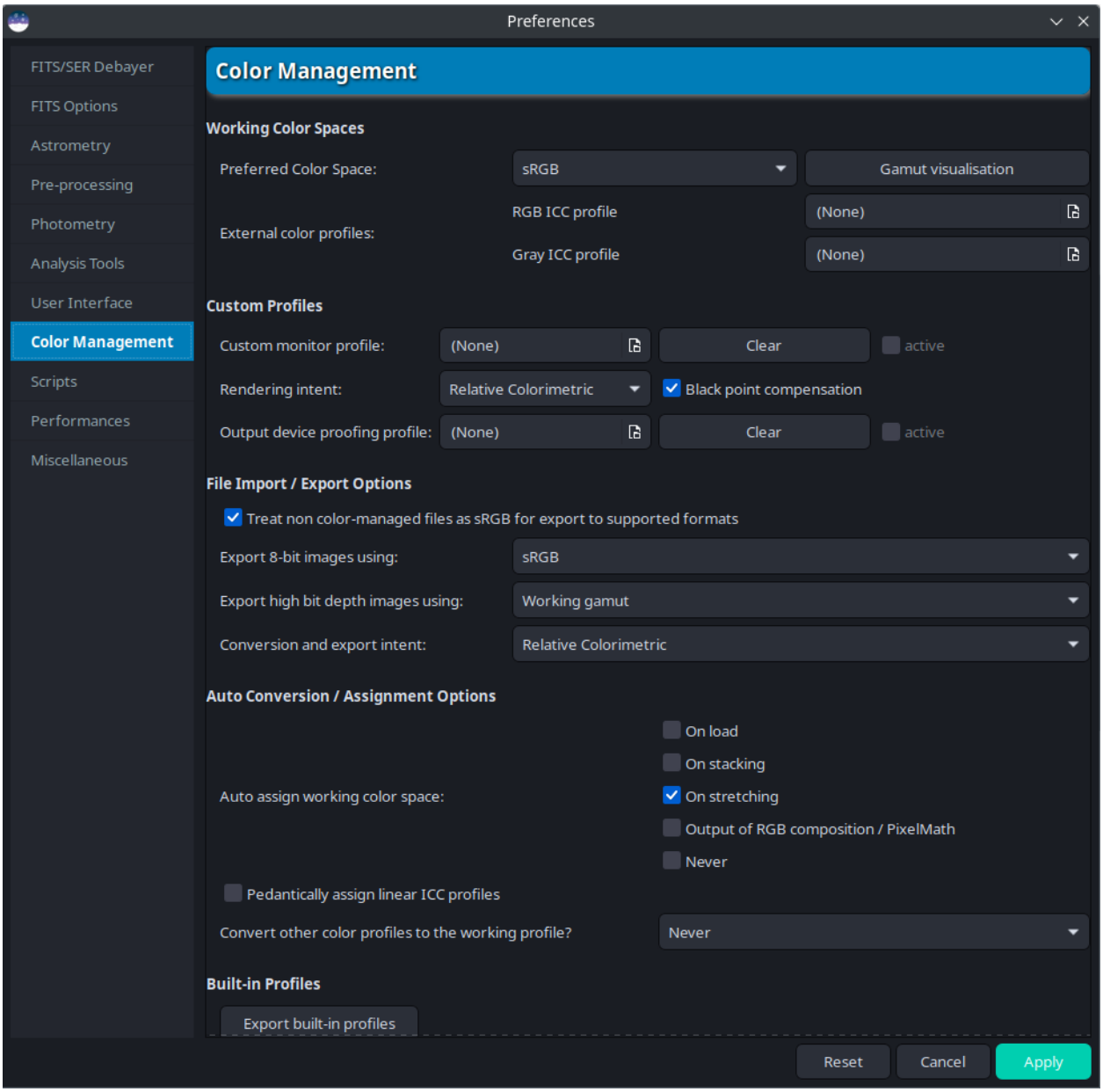
Der Bildschirm wird auf sRGB gesetzt und der bevorzugte Farbraum ist ebenfalls sRGB. Dies bedeutet, dass keine Farbtransformation zwischen den beiden notwendig ist und es daher keinen Verarbeitungsaufwand für Bilder mit dem Default-Farbprofil gibt.

Wenn Bilder mit einem zugeordneten Farbprofil geladen werden, wird dieses Profil beibehalten. Dies bedeutet, dass eine Farbraumkonvertierung notwendig ist und verursacht daher einen kleinen Umrechnungsaufwand, minimiert aber Änderungen am geladenen Bild.

Wenn Bilder kein Farbprofil haben, wird ihnen beim ersten Streckvorgang der bevorzugte Farbraum (sRGB) zugewiesen. Dies ist der Punkt, an dem Sie den linearen Anzeigemodus verwenden sollten, damit das Bild auf dem Bildschirm so erscheint, wie Sie es wünschen, und daher ist dies der Punkt, ab dem das Farbmanagement wichtig wird.

Die Bilder werden in alle 8-Bit- und 16-Bit-Formate exportiert, wobei der sRGB-Farbraum verwendet wird.

Diese Standardeinstellung ist völlig in Ordnung, wenn Ihr Hauptziel die Produktion von Bildern für das Internet oder die Anzeige auf den meisten typischen Computermonitoren ist.

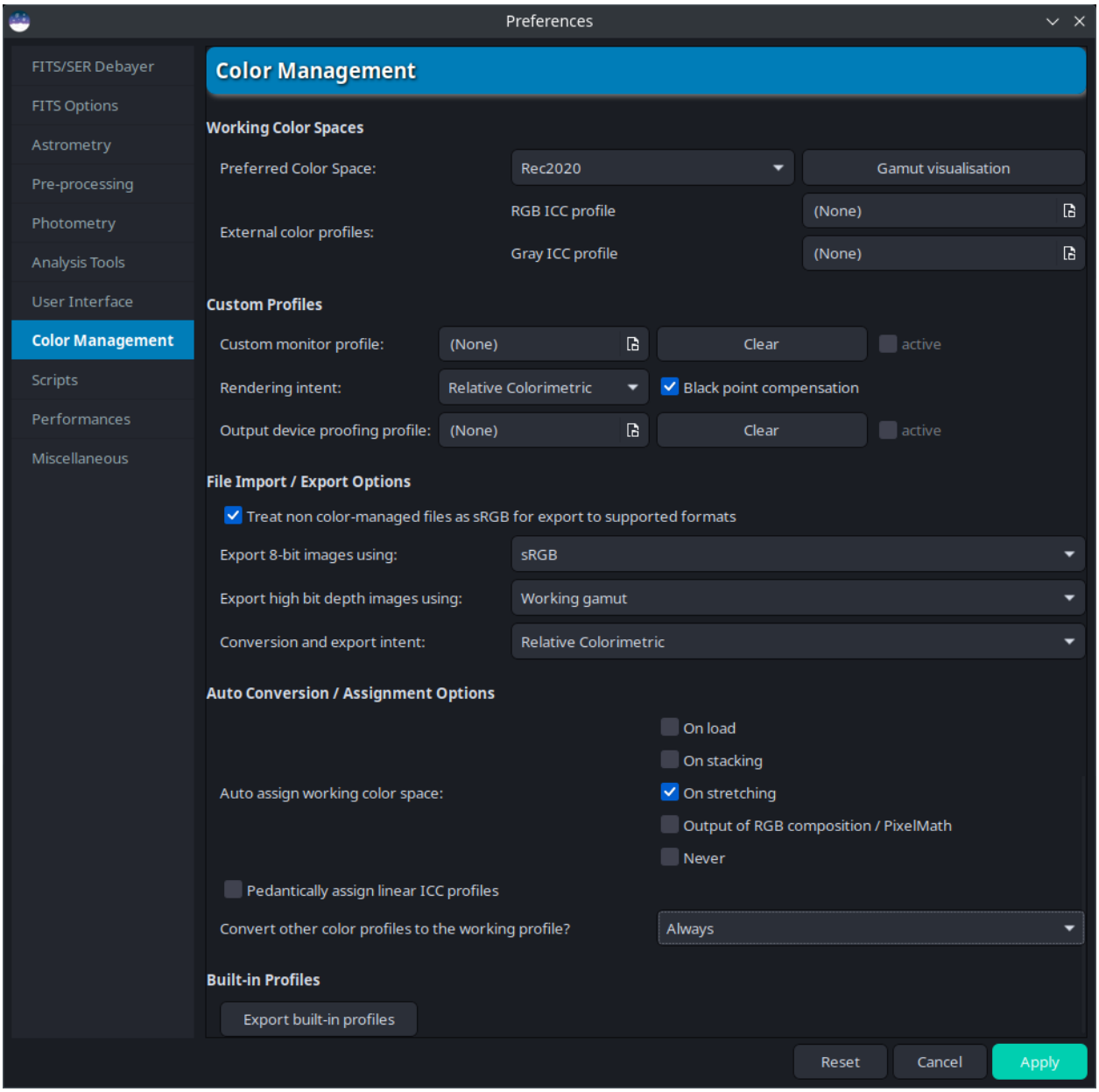


6.1.2 Einstellungen Foto-Druck

Wenn Sie planen Ausdrucke ihrer Arbeit zu erstellen, erwägen Sie folgende Einstellungen.

Der Monitor bleibt im Screenshot sRGB, da kein benutzerdefiniertes Monitorprofil eingestellt ist, aber wenn Sie Zugang zu der notwendigen Hardware haben, um Ihren Monitor zu kalibrieren, können Sie ein benutzerdefiniertes Monitorprofil einstellen, das auf Ihren Bildschirm zugeschnitten ist. (Wie das geht, liegt außerhalb des Rahmens von Siril, aber vielleicht möchten Sie sich das freie Software-Produkt ArgyllCMS ansehen).

Der bevorzugte Farbraum ist Rec2020. Dieser Wide-Gamut-Farbraum eignet sich besser für einen Workflow, der auf die Druckausgabe ausgerichtet ist, da die vom Drucker durchgeführte Konvertierung von RGB in CMYK das CMYK-Profil weniger wahrscheinlich beschneidet.


Preferences
FITS/SER Debayer
FITS Options
Astrometry
Pre-processing
Photometry
Analysis Tools
User Interface
Color Management
Scripts
Performances
Miscellaneous
Color Management
Working Color Spaces
Preferred Color Space:
Rec2020
Gamut visualisation
External color profiles:
RGB ICC profile
(None)
Gray ICC profile
(None)
Custom Profiles
Custom monitor profile:
(None)
Clear
active
Rendering intent:
Relative Colorimetric
Black point compensation
Output device proofing profile:
(None)
Clear
active
File Import / Export Options
Treat non color-managed files as sRGB for export to supported formats
Export 8-bit images using:
sRGB
Export high bit depth images using:
Working gamut
Conversion and export intent:
Relative Colorimetric
Auto Conversion / Assignment Options
Auto assign working color space:
On load
On stacking
On stretching
Output of RGB composition / PixelMath
Never
Pedantically assign linear ICC profiles
Convert other color profiles to the working profile?
Always
Built-in Profiles
Export built-in profiles
Reset
Cancel
Apply

Wenn Bilder geladen werden, werden sie, falls sie ein zugehöriges Farbprofil haben, in Rec2020 konvertiert. Wenn Bilder kein Farbprofil haben, wird ihnen beim ersten Dehnen das bevorzugte Profil (Rec2020) zugewiesen.

6.1.3 Rendering-Absicht

In diesen beiden vorgeschlagenen Einstellungen ist die Rendering-Absicht auf "Relativ farbmetrisch" eingestellt. Diese Einstellung sollten Sie immer für die Bearbeitung Ihres Bildes verwenden, da bei der relativen Farbmetrik die Gamut-Farben wahrheitsgetreu bleiben und dies die genaueste Ansicht dessen liefert, wie Ihr Bild insgesamt aussieht.

Mit dem relativen kolorimetrischen Intent werden jedoch Farben außerhalb der Farbskala (Gamut-Bereich) abgeschnitten. Sie werden keine Details der Färbung sehen, die außerhalb der Farbskala liegen.

Daher kann es manchmal sinnvoll sein, die Rendering-Absicht auf Wahrnehmung zu ändern. Dadurch werden die Farben aus dem Arbeitsbereich nahtlos auf die Anzeige übertragen. Alle Farbdetails werden angezeigt, allerdings um den Preis eines Verlusts der Gesamtsättigung. Dies ist unvermeidlich, da der größere Farbraum auf den kleineren Farbraum gestaucht wird. Daher bietet die Wahrnehmung-Absicht keine echte Visualisierung Ihres Bildes: Um es noch einmal zu wiederholen, er sollte **nicht** für die Bearbeitung verwendet werden, **nur** um Details in Ihren Bildern zu erkunden, die außerhalb des sRGB-Farbraums liegen. Siehe die Diskussion hier <https://www.gimpusers.com/mailmsg.php?853694a0-e0d6-fd86-1582-1d890a9f90a0%40ninedegreesbelow.com>`_.

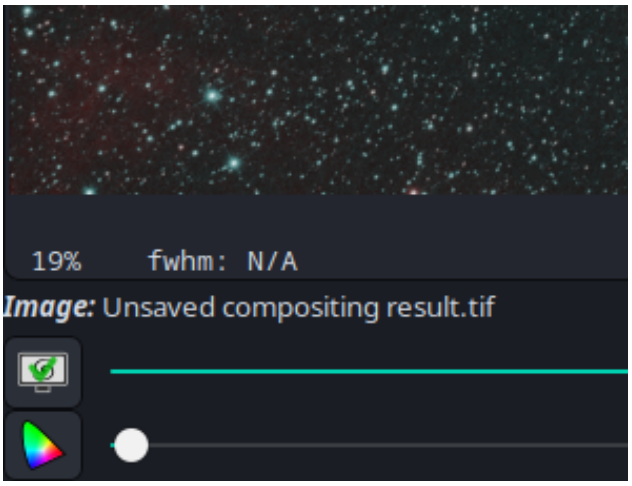
Die Bilder werden in 8-Bit-Formaten als sRGB, aber in Formaten mit hoher Bittiefe als Rec2020 exportiert (und Sie sollten Ihre endgültige Version in einem 16-Bit-Standardformat für die Druckindustrie speichern, d. h. wahrscheinlich TIFF).

6.2 Status des Farbmanagements

Der Farbmanagement-Status des geladenen Bildes wird durch das Symbol für den Farbmanagement-Status in der unteren linken Ecke des Fensters angezeigt. Das Symbol ist vollständig gesättigt, wenn das Bild farbverwaltet ist, und entsättigt, wenn das Bild nicht farbverwaltet ist. Der Tooltip liefert detailliertere Statusinformationen, wenn das Bild farbverwaltet ist: Er zeigt den Arbeitsraum und die Monitorprofile an.

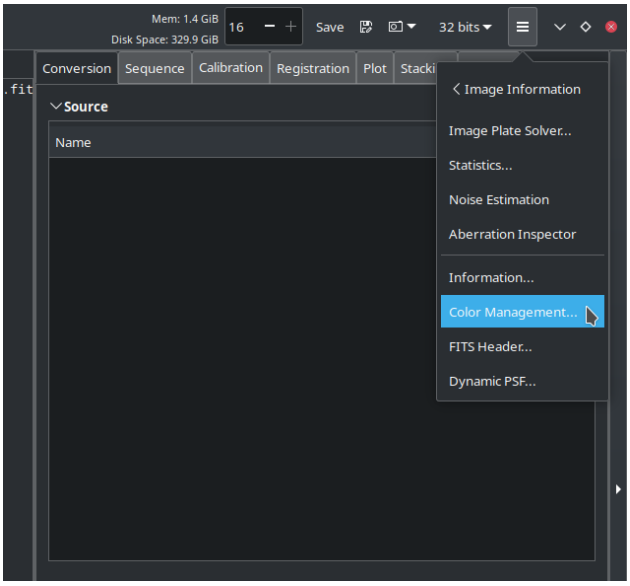
Wenn Sie mit der linken Maustaste auf die Status-Schaltfläche klicken, gelangen Sie in den Farbmanagement-Dialog. Wenn Sie mit der rechten Maustaste auf das Werkzeug klicken, wird der Anzeigemodus für die Farbbewertung nach ISO 12646 umgeschaltet. Beides wird im Folgenden näher beschrieben.

Direkt über der Schaltfläche für den Farbmanagement-Status befindet sich ein neues Menü für Bildprüfungen. Es enthält die bereits vorhandene Schnittprüfung zur Hervorhebung von Pixeln, die den Hi-Schieberegler der Benutzeroberfläche überschreiten, und fügt eine neue Gamut-Prüfung hinzu, die Pixel hervorhebt, die außerhalb des Gamuts für das Soft-Proofing-Profil in hellem Magenta liegen.



6.3 Farbverwaltungsdialog

Das Farbmanagement-Werkzeug wird durch Klicken auf das Statussymbol Farbmanagement aufgerufen. (Es kann auch über das Menü *Werkzeuge* aufgerufen werden.)



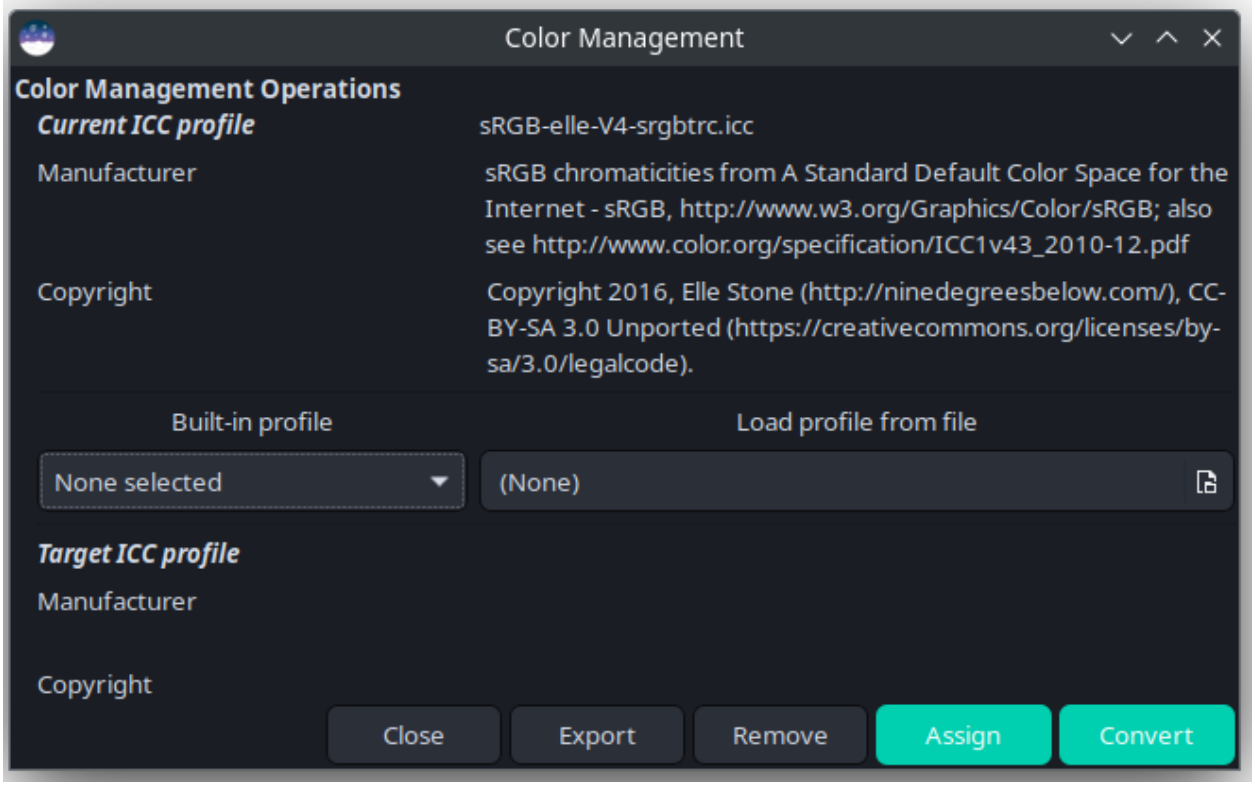
Das Werkzeug selbst wird unten gezeigt.

Im oberen Teil des Werkzeugs werden Informationen über das aktuelle ICC-Profil angezeigt, das dem Bild zugewiesen ist. Es werden die Beschreibung, die Herstellerangaben und der Copyright-Hinweis angezeigt.

Darunter befinden sich Auswahlmöglichkeiten, mit denen Sie ein anderes Profil auswählen können, entweder eines der integrierten Profile über das Dropdown-Menü auf der linken Seite oder ein beliebiges ICC-Profil, das Sie aus einer Datei laden möchten, über die Dateiauswahl auf der rechten Seite.

Warnung

Wenn Sie ein Profil aus einer Datei laden, muss der Profiltyp entweder RGB oder Grau sein, um dem geladenen Bild zu entsprechen. Die Zuweisung oder Konvertierung von Bildern in andere Farbräume wie XYZ, CIE La*b* oder CMYK wird nicht unterstützt, allerdings können Farbräume wie CIE La*b* bei Bedarf von einigen Bildoperationen intern verwendet werden.



Tipp

Sie können ein RGB-Bild in ein Grauprofil konvertieren. Das Ergebnis ist ein monochromes Bild mit der entsprechenden TRC-Transformation (Tone Response Curve) und den Anteilen der drei Kanäle R, G und B, die durch die Weißpunkte der beiden Profile bestimmt werden. Die Konvertierung eines Graubildes in ein RGB-Profil führt zu einem 3-Kanal-Bild mit der entsprechenden TRC-Transformation, wobei alle Pixelkomponenten gleich sind.

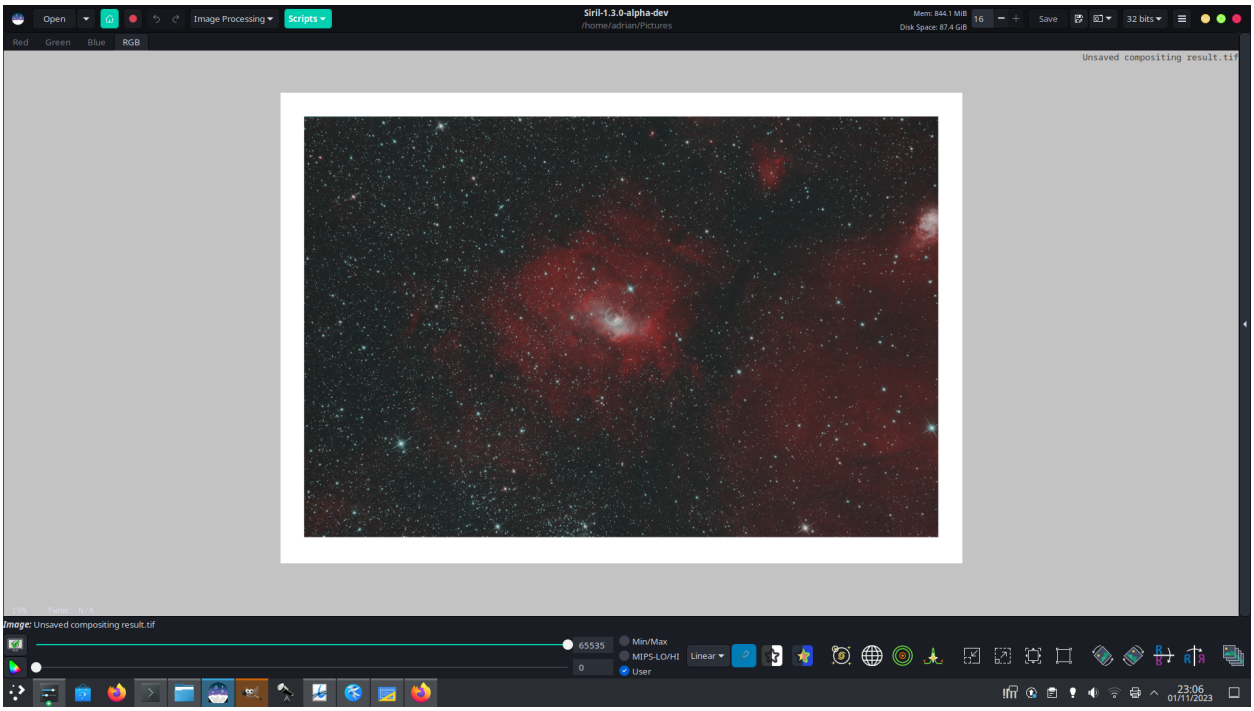
Im unteren Teil des Fensters zeigt das Tool für das Zielprofil die gleichen Informationen an wie für das ursprüngliche Profil im oberen Teil des Fensters. Dies hilft bei der Überprüfung, ob Sie das richtige Profil geladen haben.

6.3.1 Werkzeugleiste

- Die Schaltfläche *Export* exportiert einfach das Farbprofil des aktuellen Bildes in eine Datei im Arbeitsverzeichnis.
- Mit der Schaltfläche *Löschen* können Sie das Farbprofil des Bildes entfernen.
- Die Schaltfläche *Anwenden* weist dem Bild das ausgewählte ICC-Profil zu ohne eine Farbraumtransformation durchzuführen. Dies ist hilfreich, wenn einem Bild ein falsches Farbprofil zugeordnet ist.
- Die Schaltfläche *Konvertieren* konvertiert das Bild in das ausgewählte ICC-Farbraumprofil und weist dem Bild das Farbprofil zu. Dies ist hilfreich, wenn sie ein Bild aus seinem aktuellen Farbraum in einen anderen Farbraum konvertieren möchten.

Warnung

Die Verwendung der Schaltflächen *Remove*, *Assign* oder *Convert* (oder der Siril-Befehle *icc_remove*, *icc_assign* oder *icc_convert_to*) löscht jede gesetzte ROI.



6.4 ISO 12646 Bildbewertungsmodus

ISO 12646 definiert optimale Betrachtungsbedingungen für die Beurteilung von Farben. Zusammenfassend lässt sich sagen, dass die besten Betrachtungsbedingungen für eine optimale Bewertung der Farben in Ihrem Bild ein einheitlicher neutralgrauer Hintergrund mit der Bezeichnung D50 ist. Technisch gesehen sollte sich dies sogar auf die Wände in Ihrem Zimmer erstrecken, aber das kann Siril nicht kontrollieren! Siril bietet jedoch einen Bildbewertungsmodus, der sich den Empfehlungen der ISO 12646 annähert. Er ist verfügbar, wenn Sie mit der rechten Maustaste auf die Schaltfläche "Farbmanagement" unten links auf dem Bildschirm klicken, und setzt nicht voraus, dass das Bild farbverwaltet ist, um ihn zu verwenden.

Dieser Modus ist für eine abschließende Prüfung der Farben Ihres Bildes gedacht. Daher wird der Vorschaumodus auf linear mit den Schiebereglern auf 0 und 65535 (voller Bereich) eingestellt. Außerdem wird das Bedienfeld ausgeblendet und der Zoom so eingestellt, dass das gesamte Bild sichtbar und zentriert ist und von einem breiten Rand umgeben ist. Außerdem wird ein moderater weißer Rand um das Bild gelegt, um eine visuelle weiße Referenz zu schaffen, die von einem Hintergrund in D50-Grau umgeben ist.

Idealerweise sollten Sie auch ein neutrales graues GTK-Thema ihrer Benutzeroberfläche wählen, wie z.B. das ausgezeichnete Equilux Thema.

Dieser Modus deaktiviert sich automatisch, wenn der Zoom verändert oder das Panel wieder sichtbar gemacht wird. Er kann durch einen zweiten Rechtsklick auf die Schaltfläche Farbmanagement ausgeschaltet werden. Die Ansicht kehrt dann zu den vorherigen Zoom-Einstellungen und dem Status des Bedienfelds zurück.

6.5 Befehle

Die meisten Siril-Befehle greifen nicht in das Farbmanagement ein, verhindern aber nicht, dass vorhandene Farbprofile in einem Bild erhalten bleiben.

Die Ausnahmen sind drei spezifische Farbmanagement-Befehle, die zum Zuweisen, Konvertieren oder Löschen des ICC-Profils in einem Bild verwendet werden können.

Siril Kommandozeile

```
icc_assign profile
```

Weist dem aktuell geladenen Bild das im Argument angebenene ICC-Profil zu.
Eines der folgenden speziellen Argumente kann angegeben werden, um die jeweiligen eingebauten Profile zu verwenden: **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **working**, um das Arbeits-Mono- oder RGB-Farbprofil festzulegen (nur für Mono-Bilder), **linear**, oder es kann der Pfad zu einer ICC-Profildatei angegeben werden. Wenn ein eingebautes Profil angegeben wird und ein Schwarzweißbild geladen ist, wird das Grauprofil mit der entsprechenden TRC verwendet

Siril Kommandozeile

```
icc_convert_to profile [intent]
```

Konvertiert das aktuell geladenen Bild zum angegebenen ICC-Profil.
Eines der folgenden speziellen Argumente kann angegeben werden, um die jeweiligen eingebauten Profile zu verwenden: **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **graysrgb**, **grayrec2020**, **graylinear* oder **working**, um das Arbeits-Mono- oder RGB-Farbprofil festzulegen (nur für Mono-Bilder), **linear**, oder es kann der Pfad zu einer ICC-Profildatei angegeben werden. Wenn ein eingebautes Profil angegeben wird und ein Schwarzweißbild geladen ist, wird das Grauprofil mit der entsprechenden TRC verwendet.

Ein zweites Argument kann angegeben werden um die Absicht der Farbtransformation anzugeben. Die sollte eine der folgenden sein: **perceptual** (Wahrnehmung), **relative** (für relativ farbmetrisch), **saturation** (Sättigung) or **absolute** (für absolut farbmetrisch)

Siril Kommandozeile

```
icc_remove
```

Entfernt das ICC-Profil aus dem aktuellen Bild, sofern vorhanden

6.6 Farbkonvertierungsmatrix

6.6.1 Einführung

ICC-Profile decken nicht alles ab, was ein Benutzer in Bezug auf die Farbmanipulation tun möchte, daher werden auch zusätzliche Tools bereitgestellt. Natürlich ist *pixel math* ein leistungsstarkes Allzweckwerkzeug zur Manipulation von Pixeln, aber ein häufiger Anwendungsfall ist die Anwendung einer Farbkorrekturmatrix auf Daten, beispielsweise zur manuellen Konvertierung von Kamerachromatizitäten .

Warnung

Die Anwendung von CCMs ist eine fortgeschrittene Technik. Die Beschreibung von Techniken, die die Verwendung von CCMs beinhalten, liegt außerhalb des Rahmens der Siril-Dokumentation. Sie sollten verstehen, wie CCMs funktionieren und wie Sie sie in Ihrem Arbeitsablauf anwenden können, um dieses Tool erfolgreich zu nutzen.

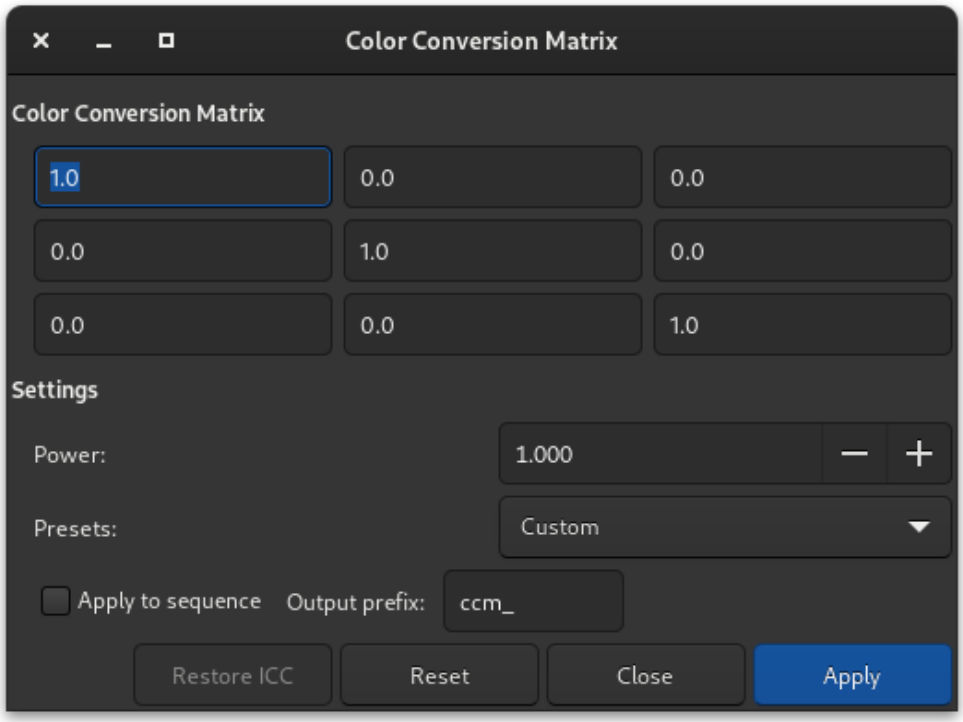


Abb. 1: Dialog Farbkonvertierungs-Matrixen

6.6.2 Farbkonvertierungs-Matrixen-Werkzeug

Das Werkzeug *Werkzeuge-> Farbkonvertierungs-Matrix* ermöglicht die direkte Anwendung einer Farbkonversionsmatrix (CCM) auf Pixel. Die CCM wird durch 9 Elemente spezifiziert:

$$\begin{pmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{pmatrix}$$

Mehrere gängige Voreinstellungen sind in einem Dropdown-Menü verfügbar. Das Werkzeug bietet zusätzlich die Möglichkeit, um einen γ-Faktor zu skalieren.

Dies wird wie folgt auf Pixel angewandt:

$$r' = (m_{00} \cdot r + m_{01} \cdot g + m_{02} \cdot b)^{(-1/\gamma)} \tag{6.1}$$
$$g' = (m_{10} \cdot r + m_{11} \cdot g + m_{12} \cdot b)^{(-1/\gamma)} \tag{6.2}$$
$$b' = (m_{20} \cdot r + m_{21} \cdot g + m_{22} \cdot b)^{(-1/\gamma)} \tag{6.3}$$

Warnung

Wenn ein CCM auf ein Bild angewendet wird, das ein eingebettetes ICC-Profil enthält, stellt das ICC-Profil keine gültige Beschreibung der Bilddaten mehr dar. Das Profil wird daher vorübergehend deaktiviert und das Farbmanagementsymbol wird als inaktiv angezeigt. Es wird davon ausgegangen, dass Ihr Arbeitsablauf Low-Level-Farbraumtransformationen und Bildoperationen umfasst und Sie die Daten irgendwann in den durch das ICC-Profil beschriebenen Farbraum zurücktransformieren werden. An diesem Punkt kann das ICC-Profil im unteren Teil des Dialogs wieder aktiviert werden. Wenn Ihr Arbeitsablauf jedoch eine manuelle Konvertierung des Bildes in einen anderen endgültigen Farbraum vorsieht, müssen Sie das Ziel-ICC-Profil über das Dialogfeld *Farbmanagement* anwenden.

Beachten Sie, dass dies nicht für den Befehlszeilenbefehl *ccm* gilt. Laut Richtlinie interagieren Siril-Befehle nicht mit ICC-Profilen, daher deaktiviert der Befehl *ccm* ein an ein Bild angehängtes ICC-Profil nicht: Es liegt in Ihrer Verantwortung, dies mithilfe von icc_remove Befehl, falls erforderlich.

Dieser Vorgang kann auf Sequenzen angewendet werden. Öffnen Sie eine Sequenz und bereiten Sie die Einstellungen vor, die Sie verwenden möchten. Aktivieren Sie dann die Schaltfläche *Auf Sequenz anwenden* und definieren Sie das Ausgabepräfix der neuen Sequenz (standardmäßig *ccm_*).

Siril Kommandozeile

```
ccm m00 m01 m02 m10 m11 m12 m20 m21 m22 [gamma]
```

Wendet eine Farbkonvertierungsmatrix auf das aktuelle Bild an.

Es gibt 9 obligatorische Argumente, die den 9 Matrixelementen entsprechen:

m00, m01, m02
m10, m11, m12
m20, m21, m22

Ein zusätzliches zehntes Argument **[gamma]** kann bereitgestellt werden: Wenn es weggelassen wird, wird standardmäßig 1,0 verwendet.

Diese werden nach folgenden Formeln auf jedes Pixel angewendet:

r' = (m00 * r + m01 * g + m02 * b)^(-1/gamma)
g' = (m10 * r + m11 * g + m12 * b)^(-1/gamma)
b' = (m20 * r + m21 * g + m22 * b)^(-1/gamma)

Siril Kommandozeile

```
seqccm sequencename [-prefix=]
```

Gleicher Befehl wie CCM, aber für die Sequenz **sequencename**. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Name der Ausgabesequenz beginnt mit dem Präfix "ccm", sofern nicht mit der Option **-prefix=** etwas anderes

angegeben wurde

Links: *ccm*

6.7 Theorie der Farbverwaltung

6.7.1 Einführung

Farbmanagement ist die Methode, mit der sichergestellt wird, dass die Farben eines Bildes immer gleich aussehen, egal wie oder wo das Bild betrachtet wird. Dies geschieht mithilfe von Farbprofilen (ICC-Profile, benannt nach dem International Color Consortium). Jeder Anzeigetyp und jede Drucker-/Papierkombination hat sein eigenes Farbprofil. Das Bild hat auch einen definierten Farbraum, und durch die Umwandlung des Bildes zwischen verschiedenen Farbräumen bei der Betrachtung auf zwei verschiedenen Arten von Bildschirmen oder beim Druck stellen wir sicher, dass es immer gleich aussieht (oder zumindest so nahe wie möglich im Rahmen der Möglichkeiten des Ausgabegeräts).

6.7.2 CIE 1931 Farbraum

Der CIE 1931-Farbraum (CIE 1931) bildet alle Farben ab, die vom menschlichen Auge wahrgenommen werden können. Da das menschliche Auge drei Arten von Zäpfchen (Farbrezeptoren) besitzt, hat CIE 1931 drei Parameter (X, Y und Z). Man beachte, dass die Parameter X, Y und Z nicht direkt der Reaktion der einzelnen Zäpfchen entsprechen, aber sie ermöglichen 3 Freiheitsgrade. Weitere Einzelheiten zu diesem Farbraum finden Sie hier: https://de.wikipedia.org/wiki/CIE-Normvalenzsystem

Es handelt sich hierbei nicht um einen Farbraum, der üblicherweise zum Speichern von Bilddaten verwendet wird, aber er ist wichtig, weil er als Zwischenfarbraum verwendet wird und weil er definiert, was "echte" Farben sind (die für das menschliche Auge sichtbaren Farben) und was "imaginäre" Farben sind (die wir nicht sehen können). Die CIE 1931 kann als Hufeisenform visualisiert werden.

Beachten Sie, dass die Außenseite des Hufeisens die Farbe der reinen, monochromatischen Spektrallinien definiert. Dies wird später wichtig sein.

6.7.3 RGB-Farbräume

Unsere Arbeitsfarbräume orientieren sich im Allgemeinen an den Farbachsen Rot, Grün und Blau. Dies entspricht in etwa der Funktionsweise unserer Augen und auch den Leuchtstoffen in den Pixeln eines Bildschirms. Es gibt jedoch eine Vielzahl von RGB-Farbräumen, die sich für unterschiedliche Zwecke eignen und jeweils Vor- und Nachteile haben. Sie können die Vielfalt der Farbräume in diesem Diagramm sehen, das sie im Vergleich zu CIE 1931 zeigen.

Die R-, G- und B-Kanäle eines Bildes werden durch einen Farbraum definiert, der aus drei Primärfarben innerhalb dieses CIE 1931-Raums sowie einem Weißpunkt und einer Tonwertkurve (TRC) für jeden Kanal besteht, die das Gamma jedes Kanals festlegt. *(Tatsächlich können TRCs komplexer sein als eine einfache Gammakurve, aber die meisten sind zumindest annähernd eine Gammakurve.)* Das obige Diagramm zeigt, dass einige Farbräume wesentlich größer sind als andere: das heißt, sie können einen größeren Teil der sichtbaren Farben darstellen.

6.7.4 Grau-Farbräume

In der Astrofotografie haben wir es sehr oft mit monochromen Bildern zu tun. Diese können natürlich kein RGB-Farbprofil haben. Stattdessen erhalten sie ein Graufarbprofil. Dieses definiert die Tonwertkurve (TRC) des Bildes genau so, wie die RGB-Farbprofile die TRC der einzelnen Kanäle definieren. Unabhängig von der Art des Bildschirms oder Druckers, den Sie verwenden, können monochrome Bilder auf verschiedenen Ausgabemedien originalgetreu reproduziert werden und gleich (oder so ähnlich wie möglich) aussehen.

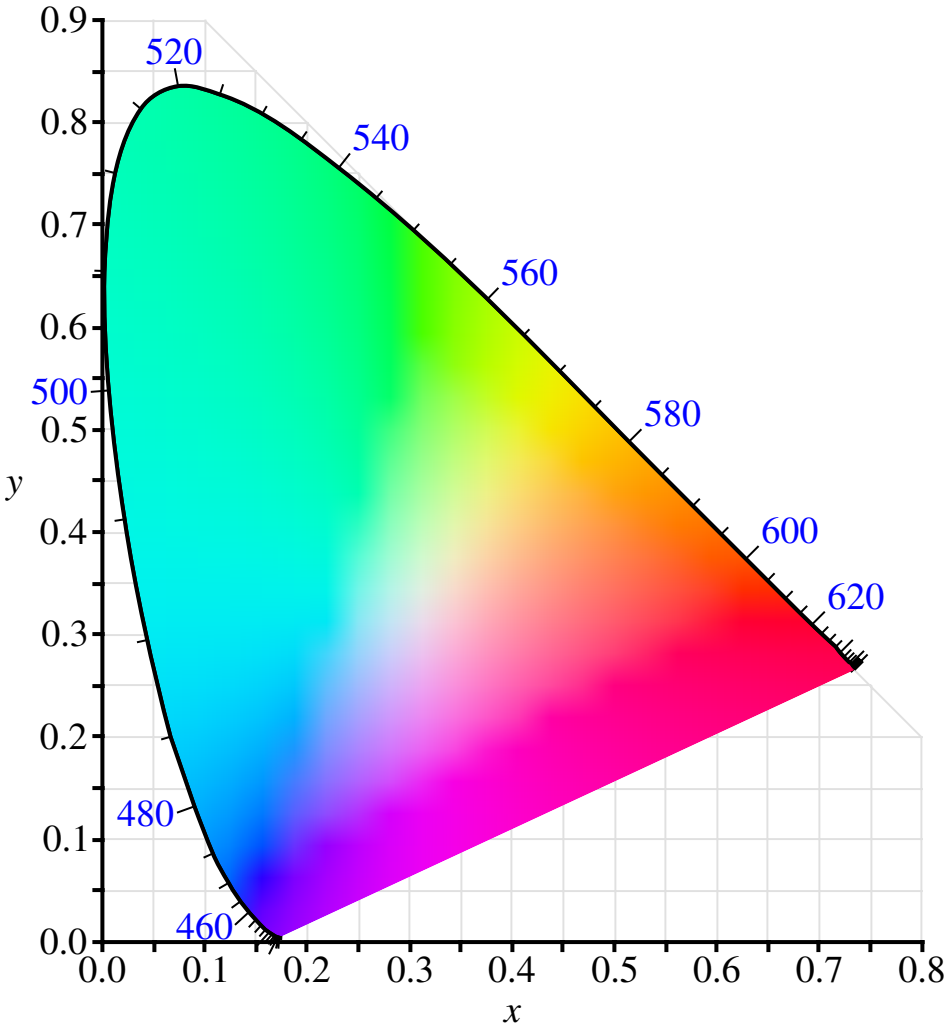


Abb. 2: Von BenRG - CIExy1931.svg, Public Domain, https://commons.wikimedia.org/w/index.php?curid=7889658

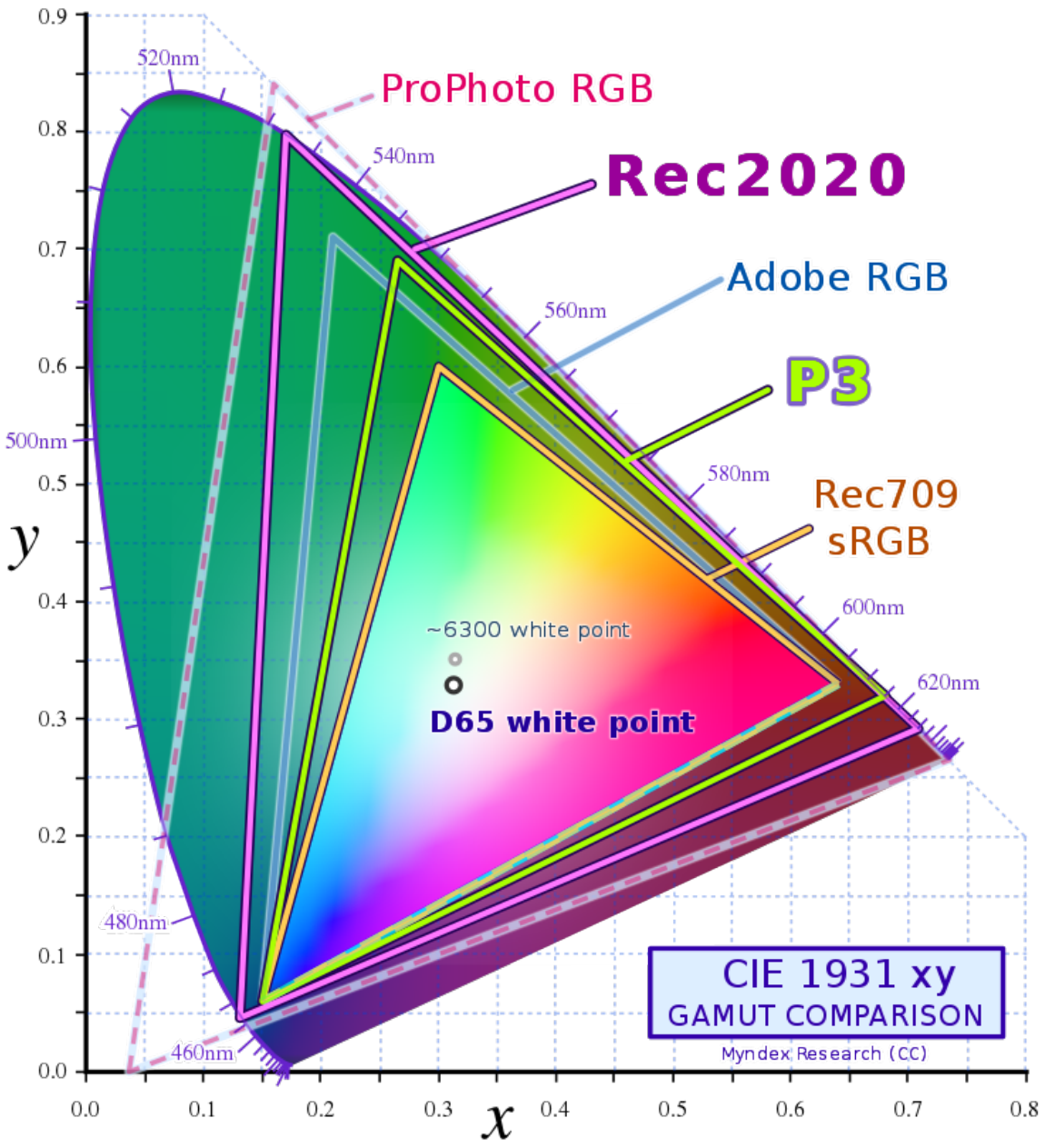


Abb. 3: Von Myndex - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=116654642

6.7.5 Bildschirm-Farbprofile

Der für Computergrafiken verwendete "Standard"-Farbraum ist traditionell sRGB. Er wurde 1996 von HP und Microsoft entwickelt und später als IEC 6166-2-1:1999 genormt. Dieser Farbraum kodiert den Farbumfang, der von den damaligen Bildschirmen erreicht werden konnte. Wie in der Abbildung zu sehen ist, kann er nur einen kleinen Teil des gesamten, von der CIE 1931 definierten sichtbaren Farbraums darstellen. Viele Monitore können jedoch auch heute noch kaum besser mehr als sRGB darstellen, und es ist der aktuelle Standardfarbraum für das World Wide Web (WWW), so dass Sie ihn als Exportprofil für jedes Bild verwenden müssen, das in Webbrowsern korrekt angezeigt werden soll, selbst wenn Sie ihn nicht für andere Zwecke verwenden wollen. Es ist auch der angenommene Farbraum für alle Bildformate, die kein Farbprofil unterstützen, und für alle Anwendungen, die kein Farbmanagement haben.

Es liegt jedoch auf der Hand, dass es andere RGB-Farbräume mit viel größeren Farbräumen gibt, z. B. Adobe RGB, Adobe ProPhoto, Rec2020. Diese können einen viel größeren Teil der in CIE1931 definierten echten Farben darstellen. Sie können auch einen viel größeren Farbraum darstellen als die meisten Monitore. Warum sollten diese also von Interesse sein?

Zum einen werden die Monitore immer besser. Wide-Gamut-Monitore sind zwar immer noch nicht weit verbreitet, aber sie werden immer häufiger und erschwinglicher. Wenn Sie Ihre Bilder auf einem Wide-Gamut-Monitor bearbeiten und betrachten, profitieren Sie von satteren Farben, da Sie einen Farbraum mit einem größeren Gamut verwenden. Einige moderne Handys können den gesamten P3-Farbraum anzeigen, der wesentlich größer ist als sRGB und eine umfangreichere Farbpalette ermöglicht.

6.7.6 Druckprofile

Druckprofile können in einigen Bereichen über sRGB hinausgehen (und in anderen Bereichen möglicherweise nicht den gesamten sRGB-Farbraum abdecken). Durch die Bearbeitung in einem Wide-Gamut-Farbraum ist diese Farbfülle - auch wenn sie in der farbumgewandelten Ausgabe auf Ihrem Bildschirm nicht dargestellt werden kann - immer noch vorhanden und wird in qualitativ hochwertigen Ausdrucken deutlich.

6.7.7 Umformungen

Der Sinn von Farbräumen besteht darin, dass ein Bild zu verschiedenen Zeitpunkten seiner Nutzung oft in Ausgabemedien mit unterschiedlichen Farbräumen erscheint. Es kann auf einem professionellen Monitor mit hohem Farbraum erstellt werden, es kann von der Öffentlichkeit auf einfachen sRGB-Monitoren betrachtet werden, und es kann auf einer Reihe verschiedener Drucker gedruckt werden. Jedes dieser Geräte ist in der Lage, unterschiedliche Farbräume wiederzugeben, aber wir wollen, dass das Bild so weit wie möglich auf allen Geräten gleich aussieht. Dies wird durch Farbraumtransformationen erreicht. Da die Farbräume unterschiedlich sind, müssen wir uns leider mit dem Problem auseinandersetzen, wie wir Farben, die aus einem anderen Farbraum transformiert wurden, in einem Farbraum darstellen können, der "außerhalb des Gamuts/Farbraums" liegt.

6.7.8 Farbraumtransformations-Absichten

Die Antwort auf das obige Problem lautet "Absichten/Intentionen". Immer wenn Sie ein Bild auf einem Gerät/Papier betrachten, das einen anderen Farbraum als Ihr Arbeitsprofil hat, wird eine Farbraumtransformation auf das Bild angewendet. Dies ist nicht so einfach wie eine einfache 2-Wege-Zuordnung zwischen zwei Koordinatensätzen. Nehmen wir an, Sie arbeiten in Rec2020. Betrachten Sie die Transformation in das Farbprofil Ihres Monitors. Bedenken Sie, dass Ihr Monitor (nehmen wir an, es handelt sich um ein sRGB-Display) nicht so viele Farben darstellen kann, wie in Rec2020 möglich sind. Der Farbraum muss also alle Farben in Rec2020 auf Farben in sRGB abbilden. Wie er dies tut, wird durch die Rendering-Absicht bestimmt.

Möglicherweise möchten Sie eine Absicht für das Rendering Ihres Bildes für den Bildschirm und eine andere Absicht für andere Zwecke wählen. Die verschiedenen von der ICC definierten und in Siril verfügbaren Rendering-Absichten werden im Folgenden beschrieben.

Tipp

- **Wahrnehmungsabsicht** Die Wahrnehmungsabsicht skaliert die Eingabefarbskala in die Ausgabefarbskala. Alle Farben werden geändert, aber die Beziehungen zwischen den Farben bleiben erhalten. Im Allgemeinen ist die Farbe im Ausgabefarbraum weniger gesättigt, aber die Sättigung im Vergleich zu anderen Farben bleibt erhalten.
- **Sättigung** Der Sättigungs-Absicht skaliert den Eingabefarbraum auf ähnliche Weise in den Ausgabefarbraum, allerdings auf eine Weise, die der Sättigung den Vorrang gibt. Dies ist in der Regel für Bilder mit lebendige, kräftigen Farben besser geeignet als für Fotografien.
- **Relativ farbmetrisch** Die relativ farbmetrische Absicht reproduziert Farben im Gamut/Farbraum akkurat, klammert jedoch Farben, die außerhalb des Farbraums liegen, auf den nächstgelegenen Punkt des Dreiecks, das die Grenzen des Zielfarbprofils darstellt.
- **Absolut farbmetrisch** Die absolut farbmetrische Absicht ist eigentlich nur für das Proofing in der Druckvorstufe von Nutzen.

6.7.9 ICC-Farbprofile und Verfügbarkeit der Absichten

Die ICC definiert die vier oben genannten Absichten (sowie einige andere, die hauptsächlich für die Farbsteuerung in Druckanwendungen verwendet werden), aber nicht alle ICC-Profile unterstützen alle Absichten. Die meisten der in Siril eingebauten Farbprofile sind Matrix Shaper Profile. Diese eignen sich sehr gut als Arbeitsfarbraumprofile, unterstützen aber meist nur die Absicht "Relativ farbmetrisch". Das ist in Ordnung, denn das ist meistens die Absicht, die wir bei der Konvertierung zwischen Farbräumen verwenden wollen. (Warum das so ist, wird weiter unten erklärt.) Wenn Sie in den Voreinstellungen eine Absicht einstellen, den Ihr ICC-Profil nicht unterstützt, greift Siril auf eine Absicht zurück, die unterstützt wird - in der Regel Relativ farbmetrisch.

6.7.10 Anzeige

Für den Bildschirm sollten Sie in der Regel Relativ farbmetrisch verwenden. Dadurch wird die Bildanzeige so konsistent wie möglich mit dem, was jeder andere mit einem farbverwalteten Bildschirm sehen wird, oder wie es aussehen wird, wenn es in einem farbverwalteten Druck-Workflow gedruckt wird.

Gelegentlich *können* Sie auf Wahrnehmungsmodus umschalten. Damit erhalten Sie keine genaue Farbdarstellung, aber es werden die relativen Farbunterschiede angezeigt, die bei der Darstellung auf Ihrem Bildschirm abgeschnitten werden. Sie können überprüfen, welche Teile des Bildes außerhalb des Anzeigebereichs liegen, indem Sie das Werkzeug Gamut Check im Menü Bildprüfung verwenden - die Pixel, die außerhalb des Anzeigebereichs liegen, werden in hellem Magenta angezeigt. Siril bietet ein integriertes sRGB-Profil mit Unterstützung für die Wahrnehmungs-Absicht und verwendet es automatisch, wenn die Wahrnehmungs-Absicht auf der Registerkarte Einstellungen ausgewählt ist (und wenn kein benutzerdefiniertes Monitorprofil aktiv ist). Wenn Sie ein benutzerdefiniertes Monitorprofil und die Wahrnehmung-Absicht verwenden möchten, müssen Sie sicherstellen, dass Ihr Profil dies unterstützt.

6.7.11 Konvertierung und Speichern

Für die Konvertierung und Speicherung von Dateien sollten Sie fast immer Relative farbmetrisch verwenden. Damit bleiben die Farben korrekt erhalten. Bei der relativen Farbmetrik werden die Farben in der Regel auf den Gamut/Farbraum des Farbraums, in den konvertiert wird, zugeschnitten, aber Sie erhalten ein konsistentes Ergebnis. Bei 32-Bit-Fließkommabildern verwendet Siril eine unbegrenzte Transformation, d. h., anstatt Farben zu beschneiden, werden negative Werte zugelassen. Sie müssen irgendwann beschnitten werden - zum Beispiel bei der Anzeige -, aber sie können gespeichert werden, und die Anwendung einer Farbtransformation in umgekehrter Richtung bringt die ursprünglichen Daten zurück. Es geht nichts verloren.

6.7.12 Soft-Proofing

Für das Softproofing benötigen Sie wahrscheinlich "relativ farbmetrisch". Die absolute Farbmetrik kann nützlich sein, um genau zu simulieren, wie Ihr Bild auf einem bestimmten Druckmedium aussehen wird, da sie den Weißpunkt eines Mediums in einem anderen simuliert (wenn Sie also proofen, wie ein Bild auf dem schmuddeligen, gelblichen Weiß von Zeitungspapier aussehen wird, dann versucht "absolut farbmetrisch", dies über Ihren Monitor zu simulieren), aber für das Softproofing eines breiteren Gamut-Farbraums gegen Ihr Anzeigeprofil sollten Sie wahrscheinlich die relative Farbmetrik verwenden.

6.8 Arbeitsablauf

6.8.1 Lineare Phase

Fast alle astrofotografischen Arbeitsabläufe beginnen mit einer linearen Phase. Das liegt daran, dass unsere Bildsensoren Photonenzähler sind und im Wesentlichen eine lineare Reaktion auf die eintreffenden Photonen zeigen. Verdoppelt man die Anzahl der eintreffenden Photonen während einer Belichtung, verdoppelt sich der Pixelwert ADU (ohne Berücksichtigung der Sensorverzerrung, die herauskalibriert wird).

Viele fotografische Standard-Workflows springen direkt in Farbräume mit einer nichtlinearen "Tonwertkurve", z. B. ist die TRC von sRGB ungefähr eine Gammafunktion $f(x) = x^{1/g}$, wobei $g = 2,2$.

Bei der Astrofotografie ist das jedoch anders. Die Darstellung dieser linearen Daten als lineare Werte in einem linearen Farbraum ist aus einer Reihe von Gründen unerlässlich. Am wichtigsten ist, dass einige Funktionen darauf angewiesen sind, dass die Daten linear sind: Die Sternerkennung entspricht Gaußschen Sternprofilen, und wenn auf die Daten eine nichtlineare TRC angewandt wurde, sind die Profile nicht mehr gaußförmig, und der Algorithmus zur Sternsuche wird schlechtere Ergebnisse liefern. StarNet wurde auf lineare Daten mit einer spezifischen Transformation (Histogramm-Stretch mit automatischen Parametern) trainiert, und auch hier führt die Eingabe von Daten, auf die bereits eine Gamma-Funktion angewendet wurde, zu schlechteren Ergebnissen. Entfaltung, Rauschunterdrückung... Viele Funktionen, die wir auf unsere Daten anwenden, setzen voraus, dass diese noch linear sind. Auch wissenschaftliche Auswertungen der Bilddaten sind in der Regel nur mit linearen Bildern möglich.

Wenn Sie möchten, können Sie in dieser Phase ein lineares Profil anwenden. Siril kann dies automatisch für Sie erledigen, wenn ein Bild geöffnet wird, wenn Sie die Einstellungen für die automatische Zuweisung festlegen und die Schaltfläche „Lineare Profile automatisch zuweisen“ aktivieren. Für monochrome Bilder wird das lineare Grauprofil angewendet. Bei RGB-Bildern (oder beim Zusammensetzen linearer monochromer Bilder zu einem Farbbild) ist das ausgewählte Profil das lineare Profil aus dem bevorzugten Satz von Farbraumprofilen, das Sie in *Einstellungen* festgelegt haben.

Wenn Sie beispielsweise den Farbraum Rec2020 gewählt haben, weist Siril Ihrem Bild das ICC-Profil Rec2020-V4-g10 zu. „Rec2020“ ist der Name der Skala; „V4“ gibt an, dass es sich um ein ICC-Profil der Version 4 handelt; und „g10“ gibt an, dass das Profil einen TRC hat, der eine Gammafunktion mit $g = 1,0$ ist (d. h. eine lineare Funktion).

Tipp

Für die meisten Benutzer reicht es aus, ein einfaches Arbeitsraumprofil zuzuweisen. Die Farbbalance kann später mithilfe der Farbkalibrierungstools korrigiert werden. Wenn Sie jedoch über eine kalibrierte Kamera verfügen, möchten Sie möglicherweise ein Eingabeprofil anwenden. Die Kamerakalibrierung und die Erstellung von Kameraprofilen ist ein fortgeschrittenes Thema und liegt außerhalb des Anwendungsbereichs von Siril, aber Sie finden die Anleitung möglicherweise in den Artikeln in Abschnitt D dieser Seite <https://ninedegreesbelow.com/photography/articles.html# Profil-Digitalkamera>`_hilfreich. Der einfachste Weg, ein Eingabeprofil anzuwenden, ist die Verwendung des Farbmanagement-Dialogs. Wählen Sie Ihr Kameraprofil mit der Dateiauswahl aus und klicken Sie auf *Zuweisen*. Das reicht eigentlich: Siril bearbeitet Ihr Bild gerne im Kameraprofil, Sie können aber auch Ihr Arbeitsprofil auswählen und auf :guilabel:„Konvertieren“ klicken, um die Daten in Ihren bevorzugten Farbraum zu konvertieren.

Ein weiterer Ansatz zur Kalibrierung Ihrer Eingabe ist die Verwendung von Farbkonvertierungsmatrizen (CCMs). Dies ist arbeitsintensiver: Wie bei Kameraprofilen müssen die CCMs außerhalb von Siril erstellt werden und können dann mit PixelMath auf Ihre linearen Daten angewendet werden.

Was Sie gerade getan haben, ist die Zuweisung der Farbraum-Primärfarben im CIE 1931 XYZ-Raum entsprechend den Punkten auf dem Dreieck, die den Rec2020-Farbraum darstellen. Obwohl die Daten linear sind, wird beim Betrachten des Bildes auf Ihrem Bildschirm die Transformation des Anzeigefarbraums unter Verwendung der von Ihnen gewählten Rendering-Absicht angewendet.

Allerdings ist es zu diesem Zeitpunkt nicht wirklich notwendig, ein Farbprofil zuzuweisen. Wenn Sie mit linearen Daten arbeiten, werden Sie höchstwahrscheinlich den Autostretch-Viewer-Modus verwenden, der ohnehin kein Farbmanagement erfordert.

Nachdem das erklärt ist, können Sie problemlos mit dem Rest Ihres linearen Arbeitsablaufs fortfahren. Kalibrierung, Registrierung, Stacking, Sternentfernung, Rauschunterdrückung ... Machen Sie es!

6.8.2 Nichtlineare Phase

Wenn Sie bereit sind, Ihr Bild zu strecken, ist es an der Zeit, noch einmal über Ihren Farbraum nachzudenken. Durch das Strecken wird das Bild von linearen Daten in nichtlineare Daten umgewandelt, sodass es für das menschliche Auge angenehm aussieht. Sie werden Ihre Daten jetzt nichtlinear machen, daher ist es vor dem Stretching ein guter Zeitpunkt, das Bild in den von Ihnen gewählten nichtlinearen Farbraum zu konvertieren, sei es sRGB oder Rec2020 oder einen anderen Farbraum Ihrer Wahl. Sie können dies entweder selbst manuell tun oder eine Einstellung für Siril festlegen, die Sie entweder dazu auffordert, den Bildfarbraum in Ihren bevorzugten Farbraum zu konvertieren, oder dies automatisch zu tun.

Sie können nun mit der Post-Stretch-Bearbeitung Ihres Bildes fortfahren und diese abschließen.

6.8.3 Bilder speichern und exportieren

Wenn es Zeit wird Ihr Bild zu speichern, haben Sie die Wahl zwischen verschiedenen Formaten:

- FITS ist das native Bildformat für Siril und ICC-Profile können in FITS-Dateien eingebettet werden. Dies erfolgt auf die gleiche Weise wie bei anderer Astrofotografie-Software, sodass mit eingebetteten ICC-Profilen gespeicherte Bilder zwischen Siril und PixInsight (und wahrscheinlich auch anderer Software, die die Einbettung von FITS ICC-Profilen unterstützt) kompatibel sind.
- TIFF und PNG sind gute Formate für den Export hochwertiger Bilder und bieten eine größere Kompatibilität mit anderer Software als FITS. Beide unterstützen die Einbettung von ICC-Profilen. Mit den Optionen im Dialogfeld „:guilabel:„Einstellungen“ können Sie festlegen, ob TIFF- und PNG-Bilder mit 8 Bit und höherer Bittiefe als sRGB, im aktuellen ICC-Profil des Bildes oder im bevorzugten Farbraum gespeichert werden, der in den Einstellungen festgelegt wurde.
- Der *de-facto*-Standard für hochkomprimierte Bilder für das WWW ist JPEG. Die Unterstützung für ICC-Profile in JPEG-Bildern variiert je nach JPEG-Bibliothek. Wenn Sie jedoch libjpegturbo Version 2.0.0 oder höher verwenden, werden ICC-Profile eingebettet. Dennoch unterstützen viele Programme von Drittanbietern überhaupt kein Farbmanagement und gehen davon aus, dass JPEGs ein Bild im sRGB-Farbraum mit dem sRGB-Standard TRC darstellen. Dies ist der Standard-Exportfarbraum für JPEGs und Sie sollten dies sorgfältig prüfen, bevor Sie sie mit einem anderen Farbprofil speichern.
- Siril unterstützt auch einige andere Bildformate: PNM/PGM, BMP usw. Diese Formate sind in der Kategorie „ICC-Profile werden nicht unterstützt“ zusammengefasst. Beim Exportieren in diese Formate werden Bilder **immer** in sRGB konvertiert.

Siril bietet eine Voreinstellung für die Export-/Konvertierungsabsicht. In so ziemlich jedem Fall sollten Sie die Einstellung bei „Relativ farbmetrisch“ belassen.

6.9 Farbraumprofile

6.9.1 Eingebaute Farbprofile

Siril enthält einen Grundbestand an Farbraumprofilen. Diese stammen aus Elle L. Stones "exzellenter Sammlung <https://github.com/ellelstone/elles_icc_profiles>`_ von freien und gut funktionierenden Farbprofilen. Ihre Website <https://ninedegreesbelow.com/>`_ enthält eine Fülle von technischen Informationen über Farbraumtheorie und -praxis und ist eine großartige Inspirationsquelle für die Anwendung der Farbverarbeitung auf Bilder. Alle Profile dort sind unter der Creative Commons Attribution-Share-Alike Unported license, version 3.0 veröffentlicht.

Die eingebauten Profile sollten eine ausreichende Auswahl an Farbräumen für die meisten Zwecke bieten: sRGB für den Webexport und als Standard-Anzeigeprofil, Rec2020 als Wide-Gamut-Profil, das der nächste Standard für hochwertige Wide-Gamut-Monitore werden könnte, und natürlich eine Reihe von Grauprofilen, die zu den RGB-Profilen passen.

- sRGB
  - Lineares Profil sRGB_elle_V4_g10
  - sRGB TRC Profil sRGB_elle_V4_srgbtrc
  - ICC sRGB-Präferenzprofil mit wahrnehmungsbezogenen Lookup-Tabellen (nur für die Anzeige)
  - sRGB TRC Profil sRGB_elle_V2_srgbtrc (zum Exportieren)
- Rec2020
  - Lineares Profil Rec2020_elle_V4_g10
  - Rec709 TRC Profil Rec2020_elle_V4_rec709
  - Rec709 TRC Profil Rec2020_elle_V2_rec709 (zum Exportieren)
- Grau
  - Lineares Profil Gray_elle_V4_g10
  - sRGB TRC Profil Gray_elle_V4_srgbtrc
  - sRGB TRC Profil Gray_elle_V2_srgbtrc (zum Exportieren)
  - Rec709 TRC Profil Gray_elle_V4_rec709
  - Rec709 TRC Profil Gray_elle_V2_rec709 (zum Exportieren)

Beachten Sie, dass die nativen TRC-Versionen der Profile in zwei Versionen vorliegen (Version 4 und Version 2). Version 4 bietet bessere Funktionen (insbesondere für Bilder mit hoher Bittiefe), einschließlich parametrischer TRCs, die eine Quantisierung vermeiden. Die Profile der Version 4 werden jedoch noch nicht allgemein unterstützt. Wenn Sie also eine Datei zur Verwendung in anderer Software exportieren, ist es am sichersten, ein V2-Profil einzubetten.

Das Grauprofil ist mit TRCs erhältlich, die den beiden integrierten RGB-Farbräumen entsprechen, und auch mit einem linearen TRC.

6.9.2 Lineare TRC-Profile

Sensordaten haben zunächst eine lineare Darstellung des Lichts. Warum also ordnen wir nicht auch ungestreckten Bildern wie einem neu gestacktem Stapel ein lineares Farbprofil zu? Nun, technisch gesehen sollten wir das vielleicht tun. Wenn Sie dies wünschen, können Sie es natürlich tun, und es wird nichts Schlimmes passieren. Es gibt auch eine Voreinstellung (Pedantisch lineare ICC-Profile zuweisen), die eine linearisierte Version des Arbeits-ICC-Profils zuweist, wenn ein Profil automatisch einem neu geladenen Bild (wenn es keine Anzeichen einer vorherigen Streckung aufweist), einem gestackten Bild oder einem neu zusammengesetzten Bild zugewiesen wird. Im Allgemeinen ist dies jedoch nicht von Vorteil. Um zu verstehen, warum das so ist, müssen wir einen grundlegenden Unterschied zwischen der Astrofotografie und der normalen Fotobearbeitung verstehen.

Bei der normalen Fotografie ist das gesamte Bild gut belichtet, mit dunkleren Schattenbereichen und helleren Lichtern. In der Rohdatenphase wird das Bild von linearem Licht in ein Arbeitsprofil umgewandelt. Dies ist jedoch umkehrbar und wird sogar häufig umgekehrt, da viele Bearbeitungsvorgänge (Farbmischung, Rauschunterdrückung usw.) am besten im linearen Bild durchgeführt werden.

In der Astrofotografie liegen jedoch alle Details sehr tief in den Schatten, und wir müssen in der Tat eine sehr starke Streckung anwenden, um ein Bild zu erzeugen, das ansprechend aussieht. Diese Streckung ist viel extremer als die leichte Änderung des Gammawerts von 1,0 auf 2,2 bei der Umstellung von linearem Licht auf sRGB, und **sie ist nicht leicht umkehrbar** - vor allem, wenn mehrere Streckungen angewendet werden. Wir müssen also immer noch die gleichen Operationen (Farbkorrektur, Rauschunterdrückung, Entfaltung usw.) am linearen Bild durchführen (und eine Menge anderer Dinge, die es in der normalen Fotografie gar nicht gibt, sollten ebenfalls an linearen Bildern durchgeführt werden: Sternmodellierung, Sternentfernung usw.)

Es ist nicht notwendig, ein lineares Farbprofil einzustellen, um diese Operationen durchzuführen - Algorithmen wie Rauschunterdrückung usw. kennen nicht einmal Farbprofile. Sie wenden sich einfach selbst auf die bereitgestellten Daten an.

Das Farbprofil kann also in jeder Phase der Bearbeitung einem Bild zugewiesen werden, obwohl es in der Regel am sinnvollsten ist, dies kurz vor dem Strecken zu tun. Wichtig ist, dass bei der Astrofotografie **alle Operationen, die an linearen Bildern vorgenommen werden müssen, vor dem Strecken** durchgeführt werden sollten. Im Allgemeinen sollten die Streckung und die abschließenden Anpassungen der Farbbalance, der Sättigung usw. so ziemlich der letzte Bearbeitungsschritt sein, der an einem Bild vorgenommen wird. Ab diesem Zeitpunkt sollten Sie das Bild in dem von Ihnen gewählten Farbraum bearbeiten, damit es bei der Betrachtung in einer beliebigen farbverwalteten Anwendung oder auf einem beliebigen Ausgabegerät ein möglichst konsistentes Bild ergibt.

6.9.3 Farbraum-Empfehlungen

Dieses Abschnitt erläutert etwas detaillierter warum Siril die eingebetteten Farbräume anbietet die es anbietet.

sRGB ist der *de facto* Standard für das Web. Wie bereits erwähnt, müssen Sie wirklich in sRGB exportieren, wenn Sie sicherstellen wollen, dass Ihre Bilder in allen Webbrowsern sowie in allen Anwendungen von Drittanbietern, die Farbmanagement unterstützen können oder auch nicht, gut aussehen. Es ist eine gute Lösung für die meisten SDR-Monitore (Monitore ohne HDR), und in Ermangelung eines geeigneten Farbmanagements sollten Bilder, die in früheren Versionen von Siril richtig aussahen, als sRGB-Bilder betrachtet werden.

Betrachten wir jedoch erneut den CIE-Farbraum 1931. Die Hufeisenform, die die Grenze des Farbraums bildet, steht für *reine Spektraltöne*. Wenn Sie eine perfekt monochromatische Lichtquelle wie einen Laser oder eine dieser natriumgelben Straßenlaternen haben, sehen Sie die Grenze von CIE 1931. Als Astrofotografen sind diese rein monochromatischen Lichtquellen für uns sehr wichtig, vor allem, wenn man Schmalbandaufnahmen macht. Diese Farben können in sRGB nicht genau dargestellt werden. Der Anteil der sichtbaren Farben, der in sRGB dargestellt werden kann, ist tatsächlich ziemlich gering. Wir können es besser machen.

Wenn also ein breiter Farbraum besser ist, warum ist dann der breite Farbraum in Siril Rec2020 integriert und nicht ein noch breiterer Farbraum? AllColorsRGB, ACES 2065 und ProPhoto RGB sind alle viel größer - ACES 2065 kann alle Teile des sichtbaren Spektrums darstellen.

Das Problem ist, dass sie dies tun, indem sie ihre Primärfarben - die Werte "100% Rot", "100% Grün" und "100% Blau" - *außerhalb* des sichtbaren Bereichs setzen. Dies ist vor allem für Schmalband-Fotografen problematisch, die den Primärfarben Filter zuweisen wollen: Sie haben dann imaginäre Farben in Ihrem Bild und müssen sich immer darauf verlassen, dass eine Farbraumtransformation das Richtige tut, um sie in etwas Sichtbares zu verwandeln. Außerdem sind einige dieser Farbräume linear, wie z. B. ACES 2065. Das ist gut für CGI-Künstler, aber nicht so gut für uns, weil die Anzeigetransformationen bei der Arbeit mit linearen Räumen langsam sind. Siril kann einige Transformationen von linearen Farbräumen über das hinaus optimieren, was lcms2 tut, aber nur, wenn die RGB-Primärfarben gleich sind, z. B. bei der Transformation von linearem sRGB nach g22 sRGB). Daher wurde Rec2020 gewählt, da es einen nichtlinearen Farbraum mit dem größten Gamut bietet, ohne imaginäre Primärfarben zu haben.

Ich empfehle Ihnen, wenn Sie nicht bereits von einem bestimmten Farbraum überzeugt sind, Rec2020 als Farbraum für die Bearbeitung sowie für den Versand von Bildern an hochwertige Druckdienstleister, die mit farbverwalteten Bildern

umgehen können, und sRGB für den Webexport zu verwenden. Es gibt aber auch andere gute Möglichkeiten, die Sie mit Hilfe von ICC-Profildateien verwenden können, z. B. Adobe RGB und ProPhoto RGB (ROMM), wenn Sie dies bevorzugen.

6.9.4 Farbprofile von Drittanbietern

Monitorprofile Siril enthält das ICC v4 sRGB-Monitorprofil mit Unterstützung für Perceptual Rendering LUTs und verwendet dieses standardmäßig. Stattdessen kann auch ein anderes Monitorprofil verwendet werden. Beachten Sie jedoch, dass nur die von einem bestimmten Profil unterstützten Renderabsichten verfügbar sind - viele weit verbreitete sRGB-Profile unterstützen nur den Renderabsicht "relativ Photometrisch".

Soft-Proofing-Profile Es gibt eine Vielzahl von Druckstandards und Papieren auf der Welt, und jede Kombination erfordert ein eigenes ICC-Profil. Siril kann nicht alle diese Profile zur Verfügung stellen. Um den Soft-Proofing-Anzeigemodus verwenden zu können, müssen Sie daher das entsprechende Soft-Proofing-ICC-Profil im Fenster "Einstellungen" angeben.

Möglicherweise möchten Sie in einem anderen als dem eingebauten Farbraum arbeiten, z. B. ProPhoto RGB. Dies wird unterstützt, aber Sie müssen selbst ICC-Profile bereitstellen. Im Fenster Einstellungen können Sie ein RGB-Profil und ein Grauprofil mit einer passenden TRC (im Wesentlichen mit demselben Gamma) festlegen. Wenn Sie beabsichtigen, Dateien in dem von Ihnen gewählten Farbraum zu exportieren, wird empfohlen, dass das von Ihnen bereitgestellte Profil ein V2-ICC-Profil ist, um maximale Kompatibilität mit anderer Software zu gewährleisten.

6.9.5 Einschränkungen von HDR-Bildschirmen

Display-Farbräume mit großem Gamut wie Rec2100 mit HLG oder PQ TRCs benötigen möglicherweise mehr als 8-Bit-Pixelpuffer, um gut angezeigt zu werden. Leider verwendet Siril die Cairo-Grafikbibliothek für die Anzeige, und Cairo kann noch keine Pixelpuffer mit mehr als 8 Bit verarbeiten. Für die meisten Benutzer sind die Auswirkungen dieser Tatsache wahrscheinlich vernachlässigbar. Wenn Sie jedoch das Glück haben, einen HDR-Bildschirm mit einer Helligkeit von mehr als 1000nit zu verwenden, der in der Lage ist, Wide-Gamut-Farbräume darzustellen, und bestimmte Kombinationen von Bild- und benutzerdefinierten Monitorprofilen von Drittanbietern verwenden, können bei bestimmten Bildern geringfügige Farbtreppenartefakte auftreten. Dabei handelt es sich lediglich um ein Anzeigeproblem, das sich nicht auf das gedruckte Erscheinungsbild Ihres Bildes auswirkt, auch nicht, wenn es in einen engeren Farbraum umgewandelt oder auf Betriebssystemen angezeigt wird, die Pixelbuffer mit hoher Bittiefe unterstützen.

Im Moment können wir nichts tun, aber wenn Cairo in Zukunft die Unterstützung für hohe Bittiefen hinzufügt, gibt es vielleicht Raum für Verbesserungen.

6.10 Soft-Proofing

6.10.1 Anzeige Soft-Proofing

Siril unterstützt Softproofing. Dabei wird im Farbraum Ihres Monitors visualisiert, wie Ihr Bild aussehen wird, wenn es auf einem bestimmten Druckstandard gedruckt wird. Um das Soft-Proofing für Ausgabegeräte zu verwenden, müssen Sie im Dialogfeld Einstellungen ein Soft-Proofing-Profil angeben.

Wenn kein Profil für das Ausgabegerät angegeben ist oder wenn es mit dem Kontrollkästchen für die Voreinstellung deaktiviert wurde, verwendet das Softproofing das Bildschirmprofil als Proofing-Ziel. Dies aktiviert die Bildprüfung *gamut* im Menü Bildprüfungen, die die Pixel anzeigt, die außerhalb des Farbumfangs Ihres Displays liegen. Wenn die Gamut-Prüfung aktiv ist, werden die Pixel außerhalb des Gamuts in grellem Magenta angezeigt.

6.10.2 Druck-Export

Siril unterstützt nicht den Export in CMYK-Formate für Druck- und Pressezwecke. Die meisten Fotodruckdienste erwarten, dass Bilder in RGB-Formaten bereitgestellt werden, und Druckertreiber sind viel besser in der Lage, RGB in eine Mischung aus den spezifischen Tinten zu konvertieren, die von jedem Drucker verwendet werden, als eine einfache Umwandlung von RGB in CMYK auf Anwendungsebene. Wenn Sie nicht genau wissen, warum Sie ein Bild in CMYK

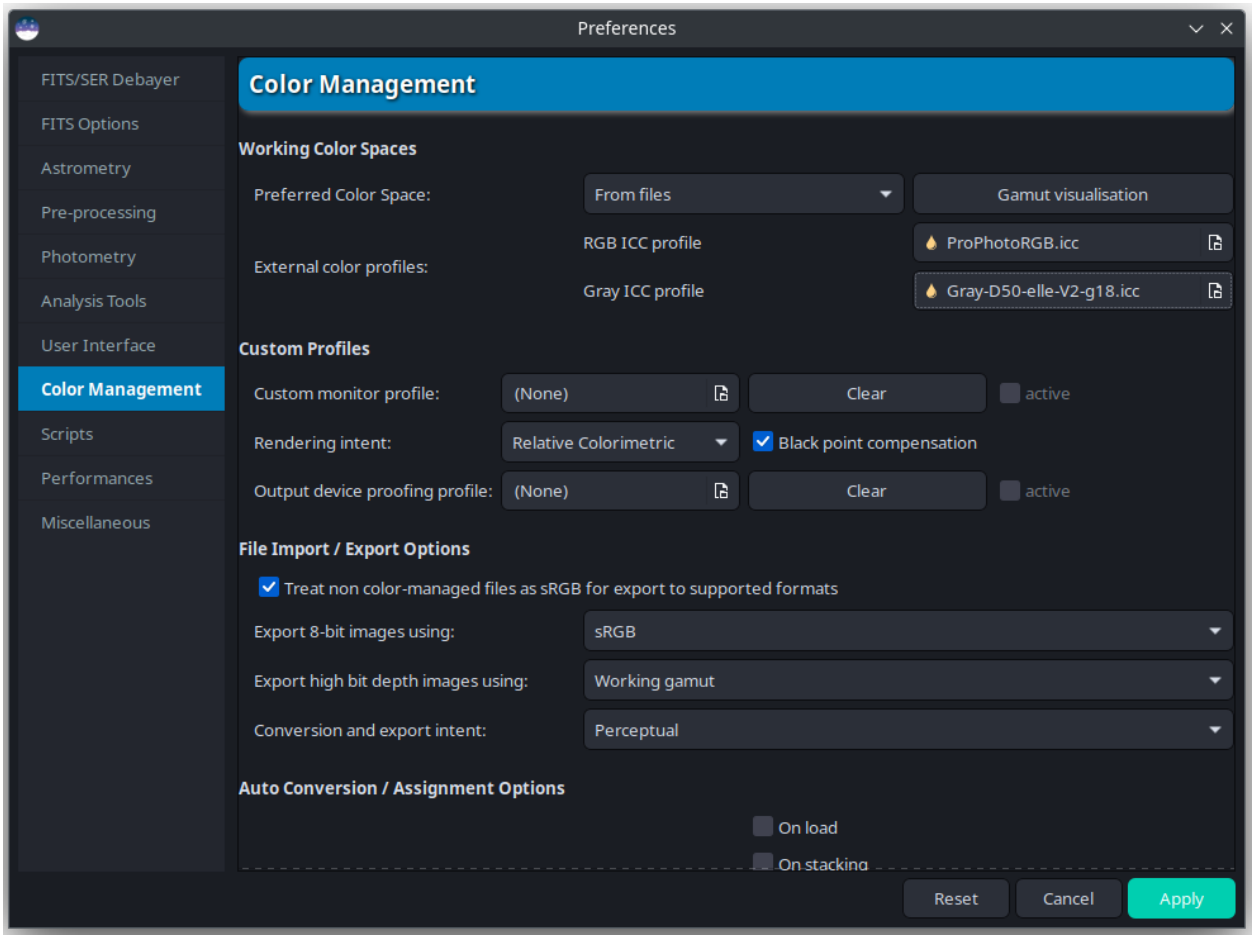


Abb. 4: Die Abbildung zeigt, wie man einen ProPhoto-Arbeitsraum mit den Profilen von Elle Stone einrichtet. Das Standard-Gamma von ProPhoto RGB (ROMM) ist 1,8, also fügen wir zusätzlich zum ProPhotoRGB-Profil das Elle Stone Gray-Profil mit Gamma = 1,8 hinzu. Sie können alle Profile von Elle Stone hier herunterladen <https://github.com/ellelstone/elles_icc_profiles>`_. Wenn Sie das ProPhotoRGB-Profil nicht haben, können Sie auch Elle Stones "LargeRGB-elle-v2-g18"-Profil verwenden, das genau dasselbe ist, außer dass sie den Begriff "ProPhoto RGB" aus möglichen urheberrechtlichen Gründen nicht verwendet hat.

konvertieren müssen, ist dies mit Sicherheit nicht erforderlich. <https://www.youtube.com/watch?v=wX3ZcAiLg-4>`_ Um jedoch die Beschneidung des Farbumfangs zu minimieren, wenn Ihr Druckdienstleister das Bild für den Druck in ein CMYK-Profil konvertiert, lohnt es sich, ein RGB-Profil mit großem Farbumfang wie das integrierte Rec2020 zu verwenden.

6.11 Änderungen durch die Einführung von Farbmanagement

Siril macht jetzt korrektes Farbmanagement.

Das bedeutet, dass es das in der Vergangenheit nicht richtig gemacht hat. Siril war in dieser Hinsicht nicht ungewöhnlich: Viele Programme bieten kein Farbmanagement, sondern gehen davon aus, dass alles in einem sRGB-Farbraum bearbeitet wird. Das ist in den meisten Fällen in Ordnung, aber es führte zu Berichten über Probleme beim Austausch von Bildern zwischen Siril und Grafikanwendungen mit ordnungsgemäßem Farbmanagement.

Einige Benutzer berichteten, dass mit Siril gespeicherte Bilder unter bestimmten Umständen in anderen Programmen anders aussahen als in Siril. Bei Bildern, die aus dem neuen farbverwalteten Siril exportiert werden, sollte dies nicht mehr vorkommen. Die Bilder werden in allen Programmen mit korrekter Farbmanagement-Implementierung einheitlich aussehen.

Wenn Sie jedoch gestreckte Bilder als FITS in einer früheren Version von Siril gespeichert haben und diese in Siril mit Farbverwaltung öffnen, können sie unter Umständen blass und verwaschen aussehen. Kein Grund zur Panik! Hier finden Sie eine Erklärung, warum das passiert und wie Sie es ganz einfach vermeiden bzw. beheben können.

6.11.1 Warum passiert das?

Wenn Siril ein FITS-Bild ohne eingebettetes ICC-Profil liest, versucht es bei der Einstellung "ICC-Profil automatisch zuweisen" herauszufinden, ob das Bild gestreckt wurde oder nicht. Das kann es nur, indem es die HISTORY-Einträge im FITS-Header überprüft. Wenn es feststellen kann, dass Streckfunktionen angewendet wurden, geht es davon aus, dass das Bild auf einem sRGB-Monitor gestreckt wurde, und wendet das entsprechende Profil an. Ältere Versionen von Siril fügten jedoch nicht immer die Historie für jeden Befehl hinzu, und andere Software kann die Historie hinzufügen oder auch nicht, aber Siril kann nicht alle Stretch-Schlüsselwörter erkennen, die von anderer Software verwendet werden, so dass es möglicherweise nicht in der Lage ist zu erkennen, dass ein Bild gestreckt wurde. Wir gehen davon aus, dass es sich bei einem FITS-Bild ohne eingebettetes ICC-Profil um **lineare** Daten handelt.

Bei Ihren gestreckten FITS-Daten handelt es sich jedoch nicht wirklich um lineare Daten. Sie wurden gestreckt, und Sie haben sie so gestreckt, dass sie im Farbraum Ihres Monitors, der wahrscheinlich ungefähr sRGB ist, richtig aussehen.

Wenn diesem ein lineares ICC-Profil zugewiesen ist, wenden die Anzeigeroutinen eine Farbraumtransformation vom linearen Raum auf das Anzeigeprofil an und strecken es damit ein zweites Mal.

6.11.2 Wie kann man das vermeiden?

Dieses Problem tritt nur auf, wenn Sie die Option "Bevorzugten Farbraum beim Laden automatisch zuweisen" aktiviert haben. Wenn diese Option deaktiviert ist, weist Siril beim Laden der Datei kein Farbprofil zu. Sie können dann über das Dialogfeld "Farbmanagement" ein Profil manuell zuweisen.

6.11.3 Wie kann man das beheben?

Wenn Sie ein Bild mit aktivierter Option "Automatisch zuweisen" geladen haben und das falsche Profil zugewiesen wurde (oder wenn das falsche Profil aus einem anderen Grund zugewiesen wurde, z. B. wenn Sie eine Datei geladen haben, die mit einer anderen Software gespeichert wurde, die ein falsches Profil zugewiesen hat), ist die Behebung des Problems ebenfalls sehr einfach. Öffnen Sie einfach das Bild, gehen Sie zum Farbmanagement-Dialog und wählen Sie "sRGB (standard sRGB TRC)" aus der Dropdown-Liste (oder "Gray (sRGB TRC)", wenn das Bild monochrom ist). Klicken Sie auf die Schaltfläche *Zuweisen* (**nicht** *Konvertieren nach*) und Siril wird das ausgewählte Farbprofil Ihrem Bild zuweisen.

Das war's. Die Anzeige sieht nun korrekt aus und wenn Sie das Bild speichern, wird das entsprechende Farbprofil eingebettet.

KAPITEL 7

Sequenzen

Sequenzen werden von Siril verwendet, um eine Reihe von zu bearbeitenden Daten darzustellen, z.B. eine Anzahl an Darks, die wir in ein Masterdark verrechnen werden. Es ist ein sehr nützliches Werkzeug für den Umgang mit einer großen Anzahl von Dateien die miteinander verknüpft oder verrechnet werden müssen.

7.1 Ein Satz aus zwei oder mehr FITS-Dateien

Siril verwendet von Haus aus 32-Bit Gleitkommazahlen oder 16-Bit Ganzzahlen ohne Vorzeichen für die Bilder im *FITS*-Format, andere Formate werden automatisch konvertiert. Um als Sequenz erkannt zu werden, müssen die Dateinamen von FITS-Bildern einem bestimmten Muster entsprechen, nämlich:

```
basename$i.[ext]
```

- **basename** kann alles sein, was aus ASCII-Zeichen besteht. Normalerweise ist es praktisch, aber nicht zwingend erforderlich, dass er mit dem Zeichen _ beginnt. Er wird als Name der Sequenz verwendet.
- **$i** ist der Index des Bildes. Er muss eine positive Zahl sein und kann mehrere führende Nullen enthalten.
- **[ext]** ist die unterstützte Erweiterung, wie in *Einstellungen* erklärt, standardmäßig wird `fit` verwendet.

Bemerkung

Die Erweiterung, die zum Auffinden von FITS-Sequenzen im aktuellen Arbeitsverzeichnis verwendet wird, entspricht der in den Einstellungen konfigurierten Erweiterung und den von Siril erstellten Dateien.

Warnung

Einige Betriebssysteme begrenzen die Anzahl der Bilder, die gleichzeitig geöffnet werden können, was für Median- oder Mean-Stacking-Methoden erforderlich ist. Unter Windows liegt die Grenze bei 2048 Bildern. Wenn Sie sehr viele Bilder haben, sollten Sie eine andere Art von Sequenz verwenden, die im Folgenden beschrieben wird.

7.2 Eine einzelne SER-Datei

SER ist ein Format, welches eine Aufnahmesequenz von mehreren zusammenhängenden Bildern in einer einzigen Datei enthält. Es ist ein eher einfaches Format, das nicht so viele Metadaten enthalten kann wie FITS-Dateien, allerdings mehr als einfache Filme und die Daten werden nicht komprimiert. SER-Dateien können nur Bilder mit 8 oder 16 Bit pro Kanal enthalten. Es gibt drei Arten von SER-Dateien, je nach Pixelinhalt: minichrom, CFA oder Farbe (3 Kanäle).

Bemerkung

Eine SER-Datei kann entweder über *Datei* und *Öffnen*, oder mit der Schaltfläche *Suche Sequenzen* geöffnet werden.

Siehe *SER* für weitere Informationen über das SER-Format und warum Filmformate wie unkomprimiertes AVI nicht für die Astronomie verwendet werden sollten.

Warnung

Bis zu einem gewissen Grad werden auch normale Filmdateien wie AVI oder andere Container unterstützt. Die Unterstützung von Filmdateien wird zugunsten von SER aufgegeben, aber es kann immer noch nützlich sein, einen Film in Siril zu öffnen, um seinen Inhalt zu untersuchen, einige Bilder zu extrahieren oder sie zu konvertieren. Einige Operationen können immer noch durchgeführt werden, allerdings langsamer als bei anderen Sequenzen, z.B. Summenstacking. Für eine vollständige Bearbeitung stoßen Sie auf Einschränkungen und Inkompatibilitäten.

7.3 Eine einzelne FITS-Datei

Bemerkung

Auch FITS-Würfel (FITS cubes) oder FITS-Sequenzen genannt, oder kurz FITSEQ in Siril.

Das FITS-Format ist ein Container für Bilder und wissenschaftliche Daten, der mehrere dieser Daten in einer einzelnen Datei enthalten kann. Wir können damit einen ganzen Satz von FITS-Bildern in einer einzigen Datei speichern, wobei der FITS-Header jedes Bildes erhalten bleibt. Es ist das Dateiformat, welches professionelle Astronomen verwenden.

Es ist einfacher, eine Datei auf der Festplatte zu verwalten als 2000, aber da es sich um eine einzelne Datei handelt, sind einige Operationen an einzelnen Bildern der Sequenz möglicherweise nicht möglich. Insbesondere ist es in Siril derzeit nicht möglich, den Header eines Einzelbildes zu ändern.

Dieses Format ist eine Alternative zu SER für eine Einzeldateisequenz, mit 32 Bit pro Kanal und voller Header-Unterstützung.

7.4 Eine Sequenz laden

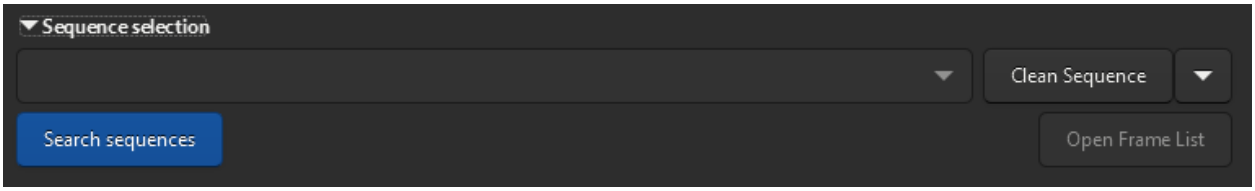
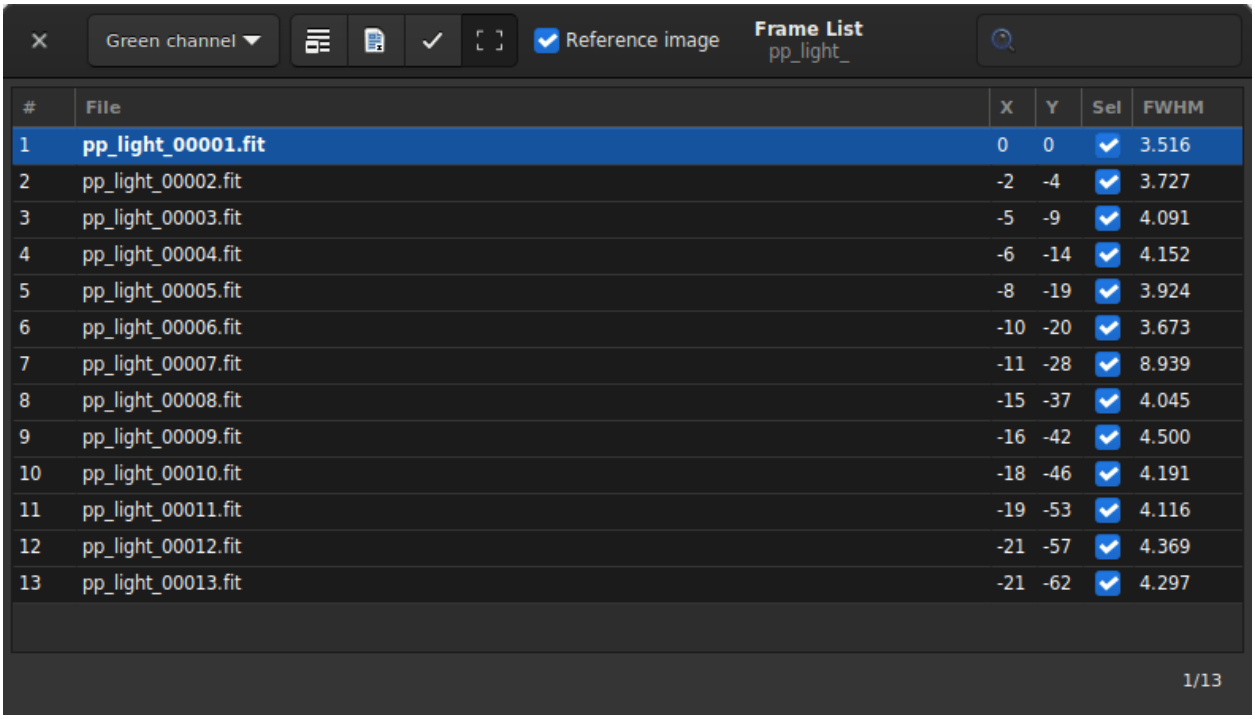


Abb. 1: Sequenzsuche und Aufräumen.

Wenn das Arbeitsverzeichnis auf den korrekten Pfad eingestellt ist, die FITS-Dateien der korrekten Nomenklatur folgen und die Erweiterung der FITS-Dateien ebenfalls korrekt eingestellt ist, klicken Sie auf die Schaltfläche *Suche Sequenzen* in der Registerkarte "Sequenzen". Es öffnet sich eine Dropdown-Liste mit allen im Ordner verfügbaren Sequenzen. Wenn nur eine gefunden wird, wird sie automatisch ausgewählt und geladen.

7.5 Bilderliste

Eine der großen Stärken von Siril ist, dass es Bildsequenzen leicht manipulieren kann. Beim Öffnen einer Sequenz wird das Referenzbild (siehe unten) angezeigt, standardmäßig ist dies das erste Bild der Sequenz. Manchmal kann es jedoch nützlich sein, einzelne Bilder einer Sequenz zu betrachten. Dies ist mit der Bilderliste möglich, der über die Symbolleiste mit der Schaltfläche oder über die Registerkarte Sequenz mit der Schaltfläche *Bilderliste öffnen* verfügbar ist.



#	File	X	Y	Sel	FWHM
1	pp_light_00001.fit	0	0	✓	3.516
2	pp_light_00002.fit	-2	-4	✓	3.727
3	pp_light_00003.fit	-5	-9	✓	4.091
4	pp_light_00004.fit	-6	-14	✓	4.152
5	pp_light_00005.fit	-8	-19	✓	3.924
6	pp_light_00006.fit	-10	-20	✓	3.673
7	pp_light_00007.fit	-11	-28	✓	8.939
8	pp_light_00008.fit	-15	-37	✓	4.045
9	pp_light_00009.fit	-16	-42	✓	4.500
10	pp_light_00010.fit	-18	-46	✓	4.191
11	pp_light_00011.fit	-19	-53	✓	4.116
12	pp_light_00012.fit	-21	-57	✓	4.369
13	pp_light_00013.fit	-21	-62	✓	4.297

Abb. 2: Bilderliste mit der Sie ein Einzelbild aus der Sequenz auswählen und anzeigen, als Referenzbild festlegen oder ausschließen können.

> **Tipp**
>
> Wenn Sie in der Bilderliste mit der Maus über ein Bild fahren, wird der ursprüngliche Dateiname angezeigt, sofern er gespeichert wurde.

Wenn Sie auf ein Bild aus der Liste klicken, wird es im Hauptbereich angezeigt, während die Sequenz als aktives Objekt für die Bearbeitung beibehalten wird. Das Werkzeug ist mehr als nur eine Bildauswahl, es kann auch dazu verwendet werden, Bilder **manuell aus der Sequenz auszuschließen** oder zu visualisieren, welche Bilder noch enthalten sind, die Werte der Bildqualität und der Verschiebung zwischen Bildern zu visualisieren (wenn sie berechnet wurden), und das Referenzbild zu ändern. Beachten sie, dass weitere Informationen zur Bildqualität auf der Registerkarte "Grafischer Plot"" angezeigt werden.

Wenn Sie ein Bild aus der Sequenz ausschließen, bedeutet das nicht, dass seine Daten endgültig gelöscht werden, es wird nur nicht für die nachfolgenden Verarbeitungsschritte verwendet, wenn dies eingestellt wird. In den meisten Fällen heißt die entsprechende Option *Nur ausgewählte Bilder verarbeiten*.

Das **Referenzbild** ist das Bild einer Sequenz, das als Ziel für die Registrierung und für die Normalisierung genutzt wird. Die übrigen Bilder werden so transformiert, dsas sie wie das Referenzbild aussehen, daher sollte es sehr sorgfältig ausgewählt werden. Glücklicherweise kann seit Siril 1.2 eine neue Registrierung in zwei Durchgängen automatisch das beste Bild der Sequenz als Referenzbild auswählen bevor mit der Bildtransformation begonnen wird.

Die Kopfzeile des Fensters bietet zahlreiche Steuerelemente für die Sequenzeigenschaften:

- Das Dropdown-Menü ermöglicht es, den Kanal zu ändern, für den die Registrierungsdaten (Qualität, Verschiebung) angezeigt werden, falls diese Daten für andere Kanäle vorhanden sind.
- Mit der ersten Schaltfläche der Symbollleiste werden alle Bilder der Sequenz als manuell ausgeschlossen gekennzeichnet.
- Die Zweite markiert alle Bilder als ausgewählt.
- Die Dritte schließt die aus der Liste ausgewählten Bilder ein oder aus (Mehrfachauswahl mit `Ctrl` oder `Shift`).
- Die letzte Schaltfläche kann deaktiviert werden, um den Rahmenindikator über registrierten Bildern auszublenden.

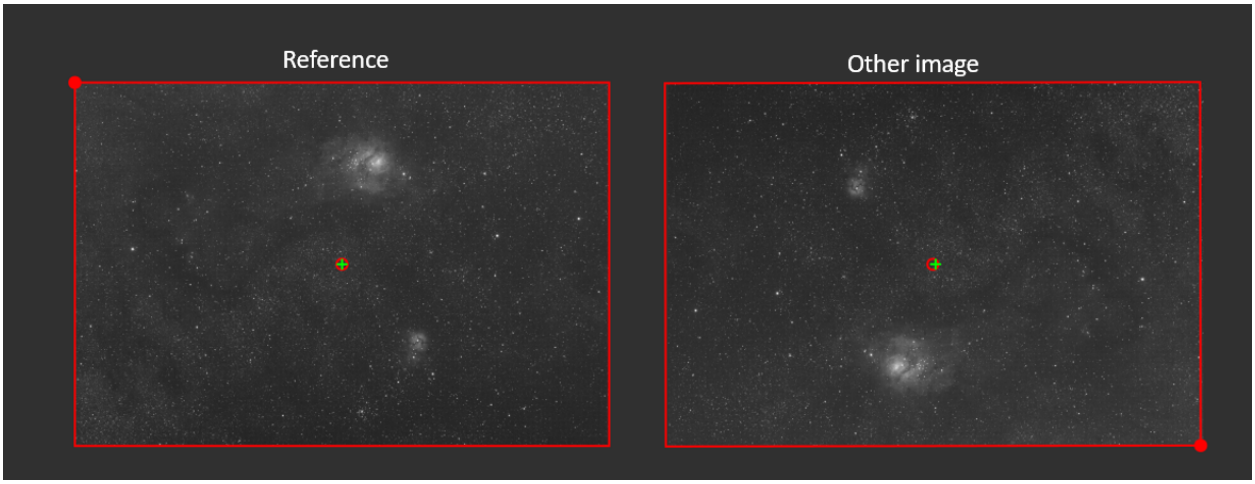


Abb. 3: Rahmenindikator
Der rote Rahmen zeigt die Spur des Referenzrahmens über jedem Bild der Sequenz. Der ausgefüllte Kreis in einer Ecke zeigt die Position der oberen linken Ecke des Referenzbilds an. Beim rechten Bild oben beispielsweise befindet sich der Rahmenindikatorkreis unten rechts, was darauf hinweist, dass zwischen diesem Bild und dem links gezeigten Referenzbild ein Meridianflip stattgefunden hat.
Der rote Kreis/das grüne Kreuz in der Nähe der Mitte zeigen jeweils die Position der Mitte des Referenzbildes und des aktuellen Bildes an. Dadurch kann die Verschiebung zwischen den beiden Bildern visualisiert werden.

Bemerkung

Seit Siril 1.3.1 ist die Farbe des Rahmenindikators unterschiedlich, je nachdem, ob die Registrierungsinformationen nur reine Verschiebungen (in Blau) oder komplexere Transformationen <preprocessing/registration:image transformations>` sind (in Rot für lineare Transformationen, in Grün, wenn Verzerrungen einbezogen werden). Bei früheren Versionen war er immer rot, unabhängig von der Transformation.

- Die Schaltfläche *Referenzbild* dient der Auswahl des Referenzbildes für die Sequenz. Alle Sequenzen müssen ein Referenzbild haben; wenn keines ausgewählt wurde, wird standardmäßig das erste Bild der Sequenz genutzt.

- Mit dem Suchfeld können Sie schließlich die Bilder über ihren Namen suchen.

Es ist auch möglich, alle Bilder zu sortieren, indem Sie auf die Spaltenüberschriften klicken. So können Sie die Bilder nach ihrer Nummer oder ihrer FWHM sortieren. Letzteres ist sehr nützlich, um sich die besten und schlechtesten Bilder anzusehen.

7.5.1 Befehle zur Verarbeitung von Sequenzen

Siril Kommandozeile

```
setref sequencename image_number
```

Legt das Referenzbild der im ersten Argument angegebenen Sequenz fest. **image_number** ist die fortlaufende Nummer des Bildes in der Sequenz, nicht die Nummer im Dateinamen, beginnend bei 1

Siril Kommandozeile

```
select sequencename from to
```

Dieser Befehl ermöglicht eine einfache Auswahl von vielen Bildern in der geladenen Sequenz (von **from** bis **to** eingeschlossen). Dies ist eine Auswahl für eine spätere Verarbeitung.
Siehe auch UNSELECT

Verweise: *unselect*

Siril Kommandozeile

```
unselect sequencename from to
```

Ermöglicht die einfache Abwahl von vielen Bildern in der geladenen Sequenz (von **from** bis **to** eingeschlossen).
Siehe SELECT

Verweis: *select*

7.6 Sequenzexport

Die Funktion **Sequenz exportieren** ist ein nützliches Werkzeug in Siril, mit dem Sie eine Sequenz von Bildern in verschiedenen Formaten exportieren können. Diese Funktion ist besonders nützlich, wenn Sie die Bilder unter Berücksichtigung der in der `seq`-Datei enthaltenen Registrierungsinformationen exportieren möchten, mit optionalem Zuschnitt und Normalisierung.

Mit der Sequenz-Exportfunktion können Sie eine zu exportierende Sequenz auswählen, das Dateiformat und die Komprimierungsstufe für Videoformate wählen. Die Sequenz-Exportfunktion von Siril unterstützt eine breite Palette von Bilddateiformaten, einschließlich FITS (einzelne FITS-Datei oder Sequenz-FITS-Datei), TIFF, SER, AVI, MP4 und WEBM und kann bei der Erstellung von Zeitraffern sehr nützlich sein.

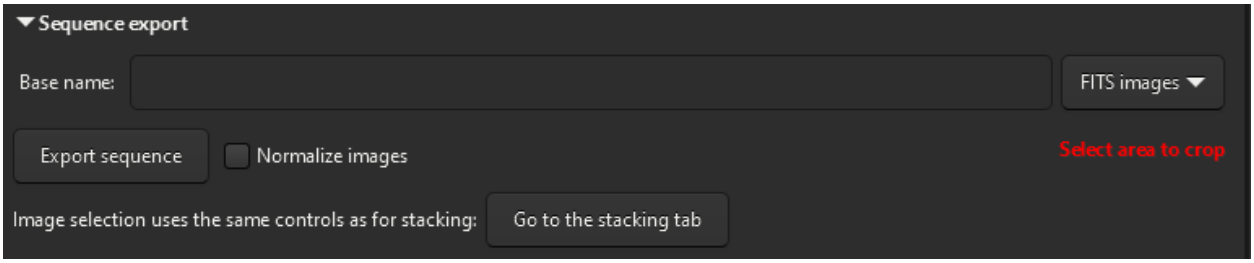


Abb. 4: Geben Sie einen Namen und ein Ausgabeformat an, um eine Sequenz zu exportieren.

Die Schaltfläche *Bilder normalisieren* ermöglicht es Ihnen, die Bilder in Bezug auf das Referenzbild zu normalisieren. Die Normalisierung entspricht derjenigen, die *beim Stacking* durchgeführt wird, mit den folgenden Einstellungen: *Additiv mit Skalierung*, *Schnellere Normalisierung* deaktiviert.

Außerdem ist es möglich, mit dem Kriterium *Bildfilterung* zu spielen, um Bilder je nach ihrer Qualität auszuschließen oder nicht. Eine Schaltfläche :guilabel:Gehe zur Registerkarte Stacking` wurde hier hinzugefügt, um einfach zu der Registerkarte zu gelangen, die die Bilder anzeigt.

7.7 Sequenzinformationen

Alle Informationen zur Sequenz, die Registrierungsinformationen, die Statistiken und die Auswahl der Einzelbilder werden in einer `.seq`-Datei gespeichert, die neben den Dateien der Sequenz abgespeichert wird. Es wird dringend empfohlen, diese Datei niemals manuell zu bearbeiten, da Siril sie kontinuierlich verändert und ein einziges falsches Zeichen das Lesen der Sequenzdatei beeinträchtigen könnte.

Eine Möglichkeit, den Inhalt dieser Sequenzdatei zu bereinigen, besteht darin, die Registerkarte **Sequenz** auszurufen und auf *Sequenz säubern* zu klicken. Die Auswahl dessen, was bereinigt werden soll, kann durch Klicken auf den kleine Pfeil daneben eingestellt werden.

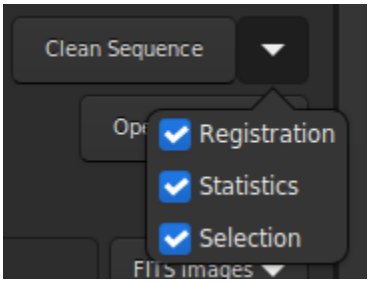


Abb. 5: Menü zur Bereinigung der Sequenzdatei.

Siril Kommandozeile

```
seqclean sequencename [-reg] [-stat] [-sel]
```

Dieser Befehl löscht Auswahl-, Registrierungs- und/oder Statistikdaten, die für die Sequenz **sequencename** gespeichert wurden.

Mit den Optionen **-reg**, **-stat** und **-sel** können Sie festlegen, dass nur die Registrierung, die Statistik und/oder die Auswahl gelöscht wird. Wenn keine Option angegeben wird, werden alle gelöscht

KAPITEL 8

Definitionen und Arbeitsablauf

Astrofotografie ist der Prozess der Aufnahme von Bildern von Himmelsobjekten. Sie umfasst mehrere Schritte, darunter die Kalibrierung und die Verarbeitung, die zwar getrennt, aber miteinander verbunden sind.

Kalibrierung ist der erste Schritt der Arbeit mit astrophotographischen Rohdaten. Er beinhaltet die Vorbereitung der Daten für die weitere Verarbeitung. Dieser Schritt umfasst in der Regel eine Dunkelstromsubtraktion (Darks), eine Flat-Field-Korrektur (Flats) und die Korrektur anderer grundlegender Probleme wie die Beseitigung heißer und kalter Pixel (Hot/Cold Pixel).

Verarbeitung bezieht sich auf die Nachbearbeitung der kalibrierten Daten, im Allgemeinen nach dem Stacking. Hier wendet der Astrofotograf verschiedene Techniken an, um das endgültige Bild zu verbessern und Details und Merkmale hervorzuheben. Dazu können Schärfung (Entfaltung/Dekonvolution), Farbkalibrierung, Rauschunterdrückung und Streckung des Bildes gehören, um die Sichtbarkeit schwacher Details zu verbessern.

Kurz gesagt, die Kalibrierung schafft die Voraussetzungen für die Verarbeitung, indem sie sicherstellt, dass die Daten in einer geeigneten Form vorliegen und von unerwünschten Signalen bereinigt sind, während es bei der Verarbeitung darum geht, das Beste aus dem Signal herauszuholen, um das endgültige Bild zu erzeugen. Beide Schritte sind in der Astrofotografie wichtig, und die Qualität des Ergebnisses hängt von den in beiden Phasen angewandten Fähigkeiten und Techniken ab.

In Siril erfolgt der Großteil der Vorverarbeitung/Kalibrierung in der Reihenfolge der Registerkarten im rechten Fensterbereich und erfordert die Verwendung von Masterdateien. Dieser Prozess kann recht einfach automatisiert werden, und die in Siril enthaltenen Skripte übernehmen diese Aufgabe. Die Bildverarbeitung wird über das spezielle Menü *Bildverarbeitung* durchgeführt. Dieser Prozess ist schwieriger zu automatisieren, da er für jedes Bild spezifisch ist und aus einer iterativen Arbeit besteht.

KAPITEL 9

Vorberechnung

Dieser Abschnitt führt Sie durch die verschiedenen Schritte zur Kalibrierung Ihrer Bilder, vom Import in Siril bis zur Erstellung Ihres gestackten Summenbildes.

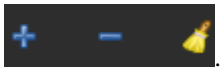
Der rechte Bereich enthält die Registerkarten, die während der Kalibrierung nützlich sind. Sie sind so konzipiert, dass sie während des gesamten Prozesses von links nach rechts verwendet werden, mit einigen Ausnahmen für die Erstellung von Master-Bildern. Diese Registerkarten sind auch über die Tastenkürzel `F1` bis `F7` zugänglich.

Die Kalibrierung (oder Vorverarbeitung) ist der Schritt, der mit der Konvertierung und dem Stacking der Bilder beginnt. Ziel ist es, alle unerwünschten Signale zu entfernen und das Rauschen auf allen Teilbildern zu reduzieren.

9.1 Umrechnung

Siril unterstützt nativ das FITS 32-Bit Format sowie das SER Format. Daher muss jedes andere Dateiformat zunächst in diese Formate konvertiert werden, um benutzt werden zu können und eine *Sequenz* zu erzeugen. Die unterstützten Dateiformate sind in der Registerkarte angegeben und hängen davon ab, wie Siril kompiliert wurde.

Siril bietet eine Umrechnungs-Registerkarte, die in 2 Bereiche unterteilt ist. Im oberen Bereich können Sie die **Quelldateien** laden, die Sie konvertieren möchten.

Die Verwaltung dieser Dateien erfolgt über die Mini-Symbolleiste  .

- Mit der ersten Schaltfläche, der Schaltfläche +, können Sie alle Ausgangsdateien laden. Sie öffnet ein Dialogfenster, in dem Sie die zu konvertierenden Dateien auf Ihrem Computer auswählen können. Es werden nur die von Siril unterstützten Formate angezeigt.

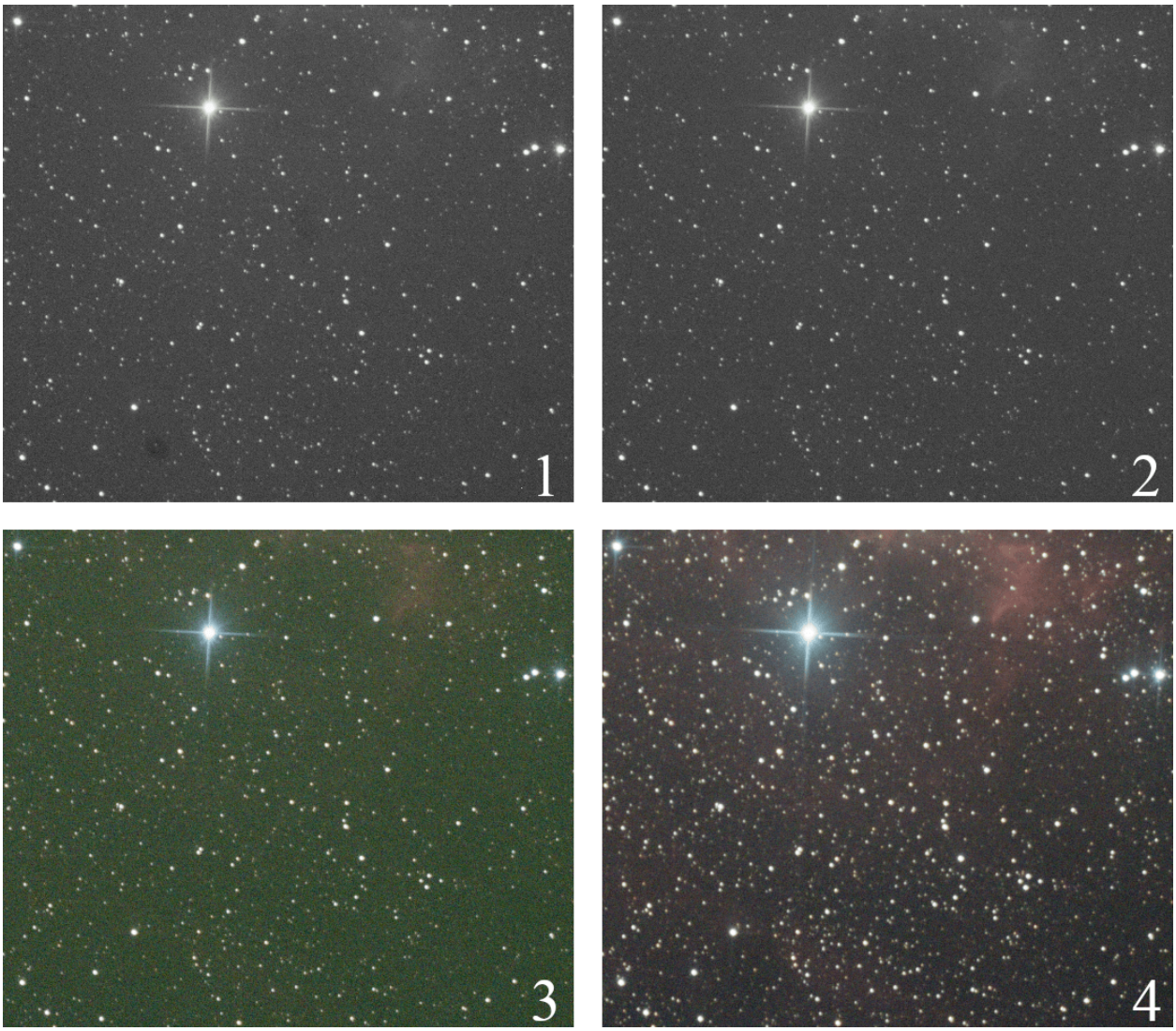


Abb. 1: **Bild 1** zeigt das Ergebnis der Konvertierung eines Rohbildes einer Digitalkamera. Man kann sichtbaren Staub erkennen, der dunklen Flecken ähnelt. **Bild 2** zeigt nach der Kalibrierung der Bilder durch die Master-Darks, -Bias und -Flats die vollständige Entfernung dieser Flecken und ein sauberes Signal. **Bild 3** ist das gleiche Bild nach dem Demosaicing (Debayern). Es zeigt die Farben und einen sehr starken Grünstich, der auf die höhere Empfindlichkeit der grünen Pixel auf den Sensoren zurückzuführen ist. **Bild 4** schließlich zeigt das Resultat des Stackings mit Kanalausgleich.

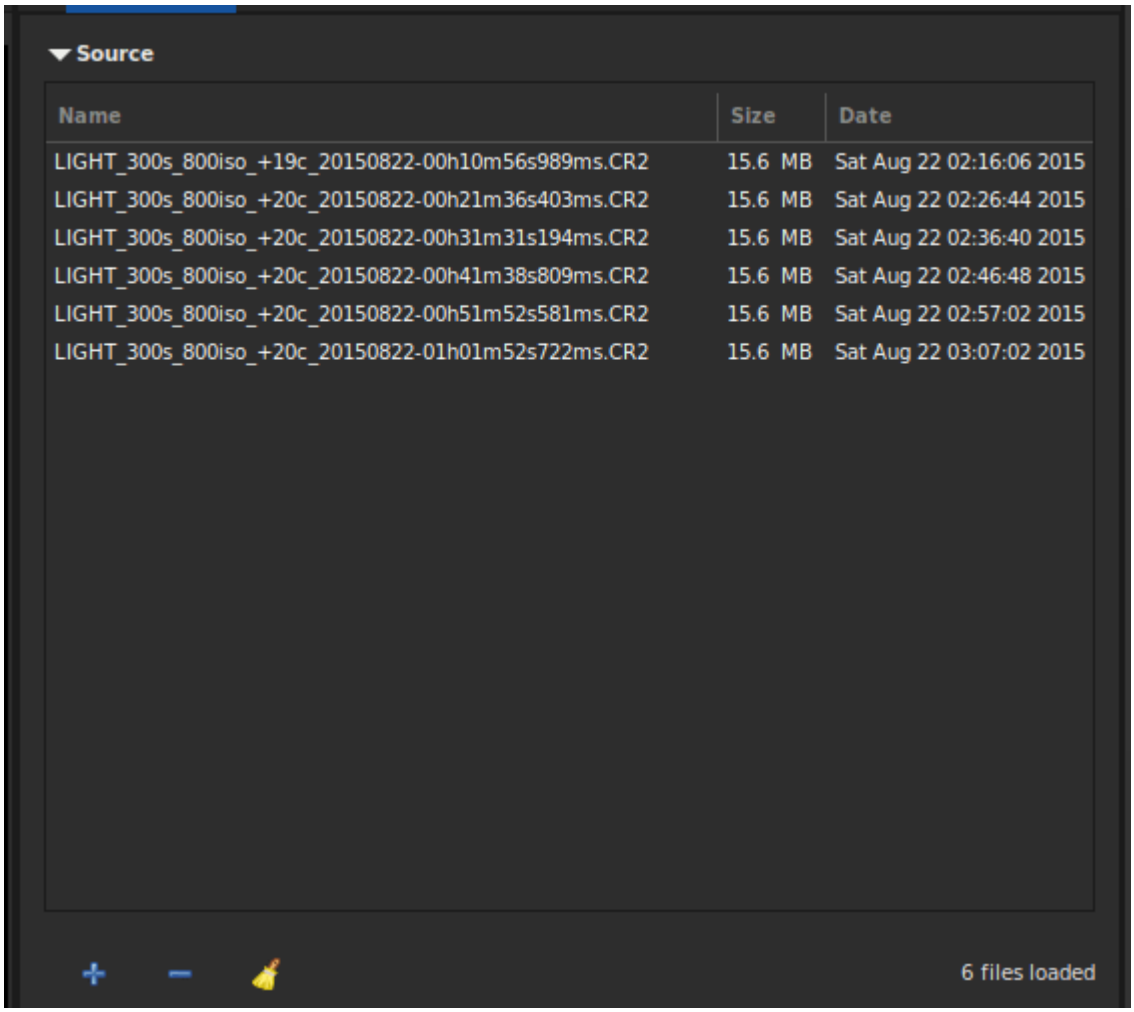


Abb. 2: Bereich "Quelle" der Registerkarte Umrechnung.

> **Tipp**
>
> Es ist auch möglich, Dateien direkt mit der Maus in den Bereich **Quellen** zu ziehen und abzulegen. Der Ablagebereich wird hervorgehoben, wenn sich die Dateien/der Mauszeiger dabei über ihm befinden.

- Die zweite Schaltfläche, die Schaltfläche -, ermöglicht das Löschen von ausgewählten Dateien. Es können mehrere Dateien gleichzeitig gelöscht werden. Sie werden nicht von der Festplatte gelöscht, sondern nur aus dem Konvertierungsbereich.
- Mit der letzten Schaltfläche können Sie alle geladenen Dateien auf einmal löschen.

Die Anzahl der geladenen und ausgewählten Dateien wird in der Statusleiste rechts neben der Symbolleiste angezeigt.

Im Abschnitt **Ziel** können Sie den Namen der Sequenz wählen, die nach der Konvertierung der Dateien erstellt wird.

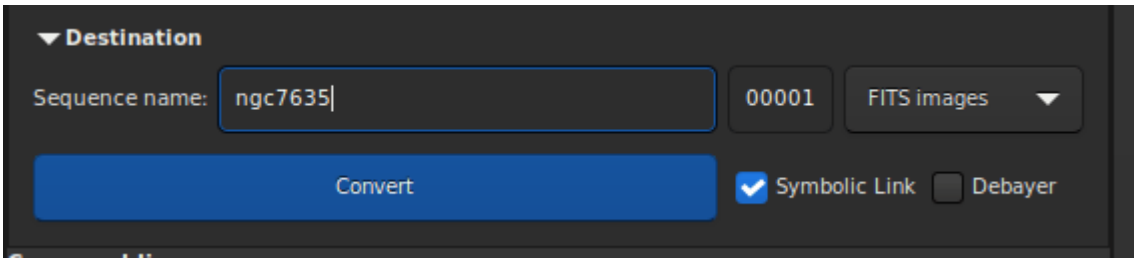


Abb. 3: Zielfeld der Registerkarte Konvertierung.

Für eine Sequenz mit dem Namen `basename` haben die konvertierten Dateien also die Form

```
basenameXXXXX.[ext]
```

Die Erweiterung ist wie in den *Einstellungen* definiert. Der `XXXXX`-Index beginnt standardmäßig bei `00001` mit dem ersten Bild, aber es ist möglich, einen anderen Startindex zu definieren. Dies kann im Falle von Mehrfachsessions (z.B. mehrere Nächte), die sich dieselben Masterdateien teilen, nützlich sein. Es sind drei Arten von Ausgabesequenzen möglich, die aus einem Dropdown-Menü ausgewählt werden können:

- FITS Bilder
- SER Sequenz
- FITS Sequenz

Diese Dateiformate werden im Abschnitt *Sequenzen* in dieser Dokumentation erläutert.

Wenn die Eingabedateien im FITS-Format vorliegen, ist es technisch gesehen nicht erforderlich, sie zu konvertieren. Sie können dies jedoch tun, damit die Dateien umbenannt werden können, um eine Sequenz zu erstellen und in Siril verarbeitet werden zu können. Um die Festplatte nicht unnötig zu füllen, kann man dann die Option *Symbolischer Link* wählen. Diese Option erstellt einen symbolischen Link für die FITS-Dateien, anstatt sie zu kopieren. Diese Option ist daher nur verfügbar, wenn die Ausgabedateien FITS-Bilder sind.

> **Bemerkung**
>
> Wenn symbolische Links aktiviert sind, wird die Komprimierung deaktiviert.

Warnung

Unter Microsoft Windows erfordert die Verwendung von symbolischen Links die Aktivierung des Entwicklermodus in Windows.

Warnung

Wenn Sie unter GNU/Linux die Fehlermeldung **Symbolic link Error: Funktion nicht implementiert** sehen, könnte das daran liegen, dass Sie versuchen, eine symbolische Linksequenz in einem Verzeichnis auf einem Dateisystem zu erstellen, das keine symbolischen Links zulässt.

Wenn es sich bei den Ausgabeformaten um SER- oder FITS-Sequenzen handelt, wird die Option *Mehrere Sequenzen* sichtbar. Aktivieren Sie diese Option, um mehrere Sequenzdateien anstelle einer einzigen SER- oder FITS-Datei für alle Eingabeelemente zu erstellen. Verwenden Sie diese Option, wenn die Eingabeelemente (Sequenzdateien wie Filme, SER- oder FITS-Würfel) nicht die gleiche Bildgröße haben oder nicht zusammen verarbeitet werden dürfen.

Die letzte Option *Debayern* ermöglicht es dem Benutzer, die Bilder während der Konvertierung zu Debayern/Demosaiken. Diese Option sollte im Allgemeinen nicht verwendet werden, wenn es sich bei den Bildern um Bias, Darks oder Flats oder um Lights handelt, die vorverarbeitet werden sollen. Aufgrund der Berücksichtigung der Bayer-Matrix ist das RGB-Ergebnis Ihres RAW-Bildes immer ein interpoliertes Bild. Die Vorverarbeitung interpolierter Daten führt daher zu falschen Ergebnissen. Die Konvertierung von RAW-Dateien eines OSC-Sensors führt zu monochromen FITS-Bildern mit Color Filter Array (CFA). Im Gegensatz zu RGB-Bildern stellen CFA-Bilder die gesamten Sensordaten mit dem Bayer-Muster dar. Das folgende Bild zeigt einen Ausschnitt aus einem CFA-Bild. Beachten Sie, dass das Bayer-Muster (in diesem Beispiel RGGB) sichtbar ist.

Die Schaltfläche *Umwandeln* schließlich ermöglicht, wie der Name schon sagt, die Konvertierung von Dateien zu starten.

Bemerkung

Die Rohbilder digitaler Spiegelreflexkameras hängen vom Hersteller ab und sind in der Regel herstellerabhängige, geschlossene Dateiformate. Daher ist die Dekodierung solcher Dateien eine komplexe Aufgabe, die von einem speziellen Code erledigt werden muss. Bei Siril wird die Aufgabe der Konvertierung von Rohdateien von LibRaw übernommen. Wenn ein Dateiformat, in der Regel ein neueres, nicht gelesen werden kann, müssen Sie auf der LibRaw-Website nachsehen, ob es unterstützt wird. Ist dies nicht der Fall, kann die Bereitstellung einer RAW-Datei dem Entwicklungsteam helfen, dies zu ändern. Es ist jedoch auch möglich, dass die in das Siril-Paket eingebettete Version von LibRaw nicht die aktuellste Version ist. In diesem Fall müssen Sie entweder auf eine neue Version warten oder die Quellen direkt kompilieren.

9.1.1 Entsprechungsdatei

Nach jeder Konvertierung wird eine Datei mit der Endung `_conversion.txt` erstellt. Sie enthält die Entsprechung zwischen den Eingabebildern und den Bildern der bei der Konvertierung erhaltenen Sequenz.

Siril Kommandozeile

```
convert basename [-debayer] [-fitseq] [-ser] [-start=index] [-out=]
```

Konvertiert alle Bilder des aktuellen Arbeitsverzeichnisses, die in einem unterstützten Format vorliegen, in eine

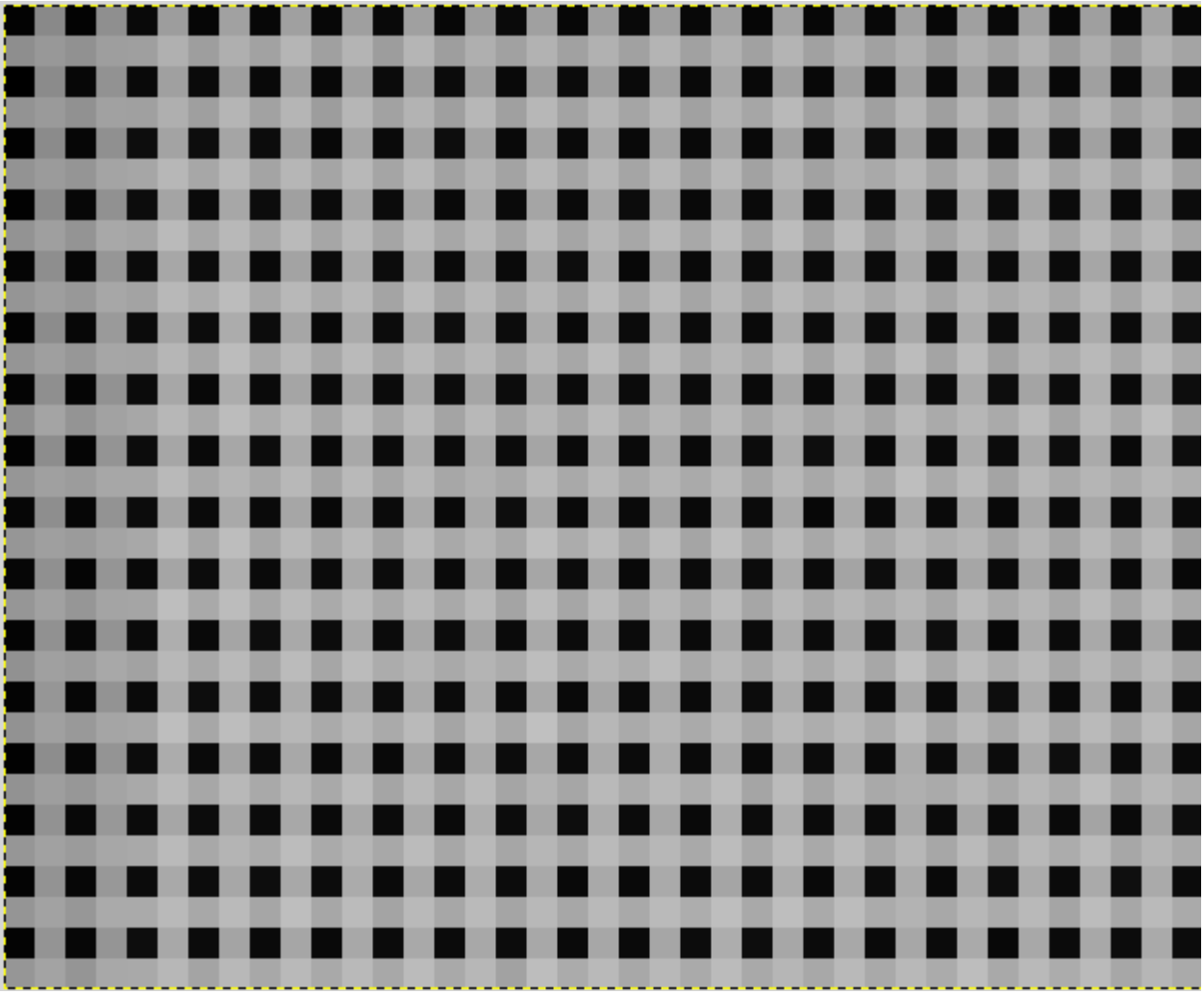


Abb. 4: Bayer-Muster auf einem CFA-Bild (Color Filter Array).

Siril-Sequenz von FITS-Bildern (mehrere Dateien) oder eine FITS-Sequenz (einzelne Datei), wenn **-fitseq** angegeben wird, oder eine SER-Sequenz (einzelne Datei), wenn **-ser** angegeben wird. Das Argument **basename** ist der Basisname der neuen Sequenz, Zahlen und die Erweiterung werden dahinter gesetzt.

Bei FITS-Bildern versucht Siril, einen symbolischen Link zu erstellen; ist dies nicht möglich, werden die Dateien kopiert. Die Option **-debayer** wendet Debayering auf CFA(Farb-)-Eingabebilder an; in diesem Fall wird kein symbolischer Link erstellt.

-start=index setzt die Startindexnummer, nützlich um eine bestehende Sequenz fortzusetzen (wird nicht mit -fitseq oder **-ser** verwendet; stellen Sie sicher, dass Sie die Ziel-.seq entfernen oder löschen, falls sie in diesem Fall existiert).

Die Option **-out=** ändert das Ausgabeverzeichnis zum angegebenen Argument.

Siehe auch CONVERTRAW und LINK

Verweise: *convertraw*, *link*

Siril Kommandozeile

```
convertraw basename [-debayer] [-fitseq] [-ser] [-start=index] [-out=]
```

Identisch mit CONVERT, konvertiert jedoch nur DSLR-RAW-Dateien, die im aktuellen Arbeitsverzeichnis gefunden werden

Verweise: *convert*

9.2 Kalibrierung

Sobald eine Sequenz geladen ist, können die Bilder kalibriert, registriert und gestackt werden. Die Kalibrierung ist ein optionaler, aber wichtiger Schritt und umfasst Bias-, Dark- und Flatframes. Die Kalibrierung einer Sequenz in Siril kann nur mit Master-Bias-, Dark- und Flatframes erfolgen, die zunächst aus Sequenzen erstellt werden müssen.

9.2.1 Masterdateien

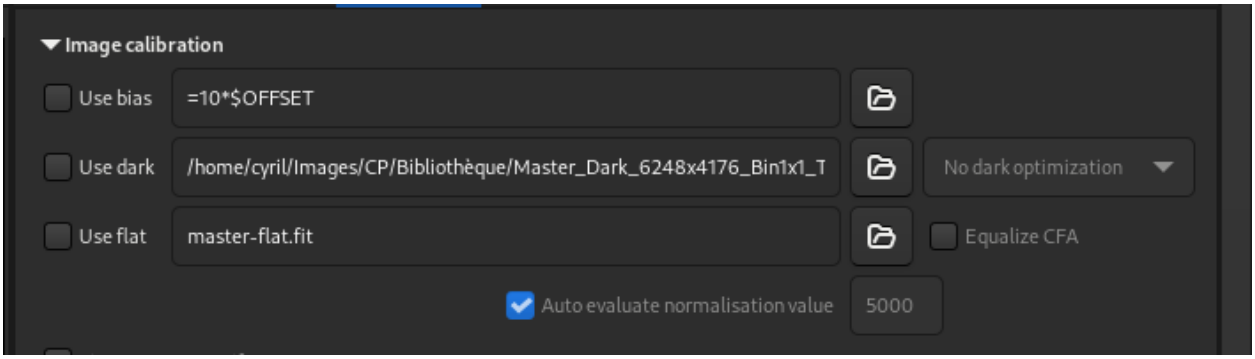


Abb. 5: Einstellungen für die Master in der Registerkarte Kalibrierung

Bias

Ich zitiere aus A Glossary of CCD terminology, um zu erklären, was ein Bias-Bild ist:

Der Bias-Pegel eines CCD-Bildes ist ein künstlich erzeugter elektronischer Offset, der sicherstellt, dass der Analog-Digital-Wandler (ADC) immer ein positives Signal erhält. Alle CCD-Daten weisen einen solchen Offset auf, der entfernt werden muss, wenn die Datenwerte wirklich repräsentativ für die pro Pixel aufgezeichneten Zählungen sein sollen.

Um ein Master-Bias in Siril zu verwenden, klicken Sie auf die Schaltfläche rechts neben dem Texteintrag und durchsuchen Sie Ihre Verzeichnisse, um das richtige Master auszuwählen. Sie können auch Master-Biasse aus einer Bibliothek verwenden, die in *Einstellungen* definiert ist.

Tipp

Das Bias-Bild muss bei verdunkeltem Kamerasensor und der kürzesten möglichen Belichtungszeit aufgenommen werden. Meist entspricht dies einer Belichtung von 1/4000s oder 1/8000s bei modernen DSLRs.

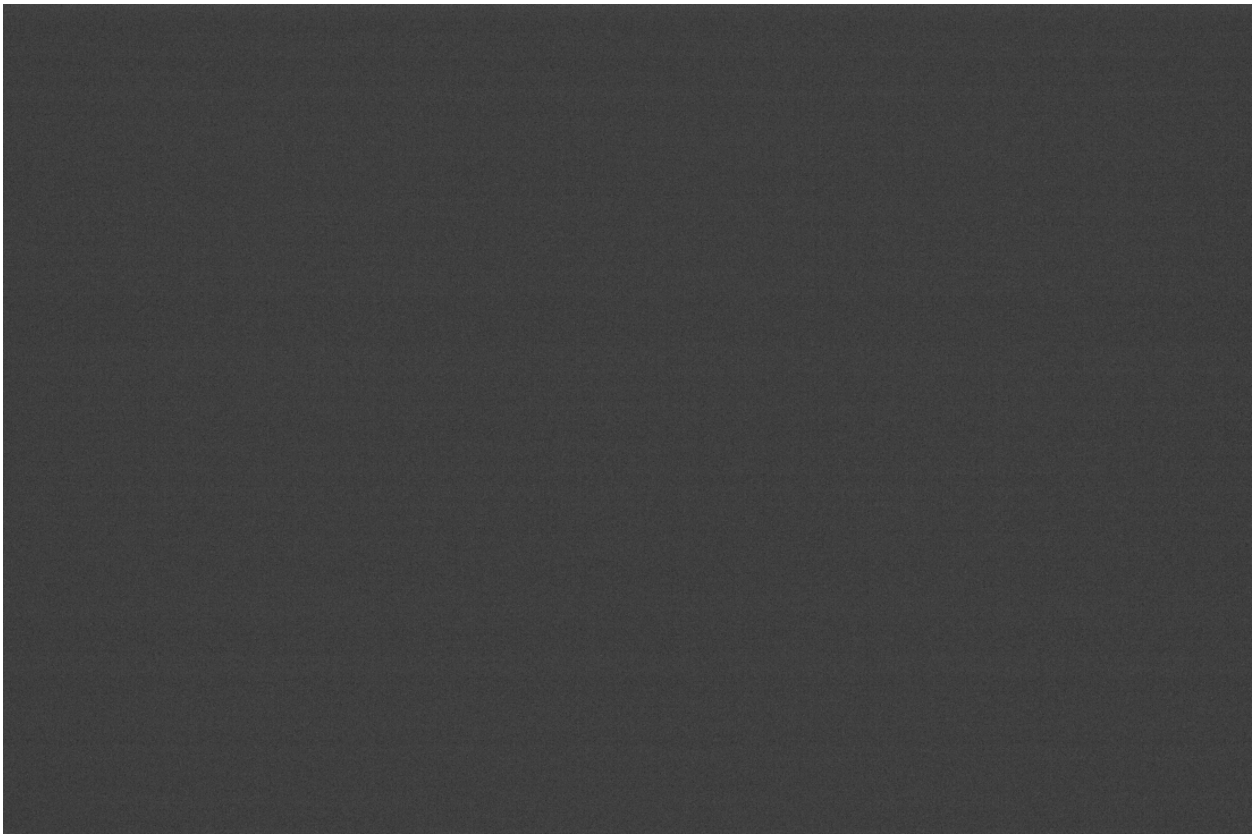


Abb. 6: Beispiel für ein Bias-Bild, das mit einer Canon EOS 1100D aufgenommen wurde. Lassen Sie sich nicht durch das leicht sichtbare Bias-Signal täuschen. Das Bild wurde automatisch gestreckt und die Unterschiede in den Signalamplituden sind stark übertrieben.

Synthetischer Bias

Da das Offset-Signal bei modernen Sensoren sehr gleichmäßig ist, empfehlen wir, es als Bild mit konstantem Offset zu verarbeiten. Dies hat den Vorteil, dass es Speicherplatz spart und das Rauschen im endgültigen Bild minimiert. Zu diesem Zweck verfügt Siril über eine Funktion, mit der dies sehr einfach zu bewerkstelligen ist.

Während der Kalibrierung Ihrer Flats können Sie, anstatt ein Masterbias anzugeben, direkt Ausdrücke in das Verzeichnisfeld eingeben, wie z. B.:

```
=2048
```

oder, wenn der FITS-Header das Schlüsselwort `OFFSET` enthält,

```
=64*$OFFSET
```

Die Zeichen = und $ sind obligatorisch. Der Wert muss in ADU angegeben werden (nicht als Float, auch wenn Sie in 32-Bit arbeiten).

Übersetzt in die Skriptsprache wird dies so geschrieben:

```
preprocess flat -bias="=64*$OFFSET"
```

Der Wert 2048 ist hier ein Beispiel für Kameras, deren Master-Bias einen Medianwert von 2048 haben würde. Im Allgemeinen ist der Wert für DSLRs proportional zu einer Wurzel aus 2. In unserem Beispiel ist $2048 = 2^{11}$.

Weitere Einzelheiten finden Sie im Tutorial Synthetische Biasse (englisch).

Darks

Darkframes enthalten das mit dem Sensor verbundene thermische Rauschen, das proportional zur Temperatur und zur Belichtungszeit ist. Daher sollten sie bei ungefähr der gleichen Temperatur wie die Lightframes gemacht werden. Dies ist der Grund, warum wir Darkframes am Ende oder in der Mitte der Aufnahmesitzung machen.

Um ein Masterdark in Siril zu verwenden, klicken Sie auf die Schaltfläche rechts neben dem Texteintrag und durchsuchen Sie Ihre Verzeichnisse, um das richtige Master auszuwählen. Sie können sogar Master-Darks aus einer Bibliothek verwenden, die in den *Einstellungen* definiert ist.

> **Tipp**
>
> Darkframes werden mit der gleichen Belichtungszeit und ISO/Gain-Einstellung wie die Lightframes des Motivs aufgenommen, aber mit verschlossenem Objektiv/Teleskop.

Dark-Optimierung

Die Option zur Dark-Optimierung ist nützlich, wenn Darks nicht unter optimalen Bedingungen aufgenommen wurden. Siril bietet zwei Methoden mit unterschiedlichen Ansätzen an, die über eine Dropdown-Liste zugänglich sind.

1. Mit der Option *Auto-evaluation* kann die Dunkelsubtraktion so optimiert werden, dass das Rauschen des resultierenden Bildes (heller Rahmen minus dunkler Rahmen) minimiert wird, indem ein Koeffizient auf den dunklen Rahmen angewendet wird.
2. Die zweite Option, *Belichtung verwenden*, basiert auf den Belichtungszeiten der Bilder, sofern diese in den FITS-Schlüsselwörtern registriert wurden.

Hier ist ein Beispiel für eine Situation, in der die Verwendung dieser Option erforderlich ist. Die Bilder wurden mit einer **FLI ProLine 4240 Kamera** aufgenommen. Das verwendete Master-Dark stammt aus einer Bibliothek und wurde mit einer Belichtung von 600s erstellt. Die einzelnen Subs hingegen haben eine Belichtungszeit von 60 Sekunden. Der Master-Dark hat eine sehr ausgeprägte und eher unansehnliche Signalsignatur: Das Vorhandensein von 4 Vorverstärkern in der Kamera ist die Ursache für ein solches Signal. Dieser Defekt ist offensichtlich auch im Galaxienbild vorhanden, und die Kalibrierung mit den Darks muss sorgfältig durchgeführt werden, um ein Bild zu erhalten, das frei von allen Defekten ist.

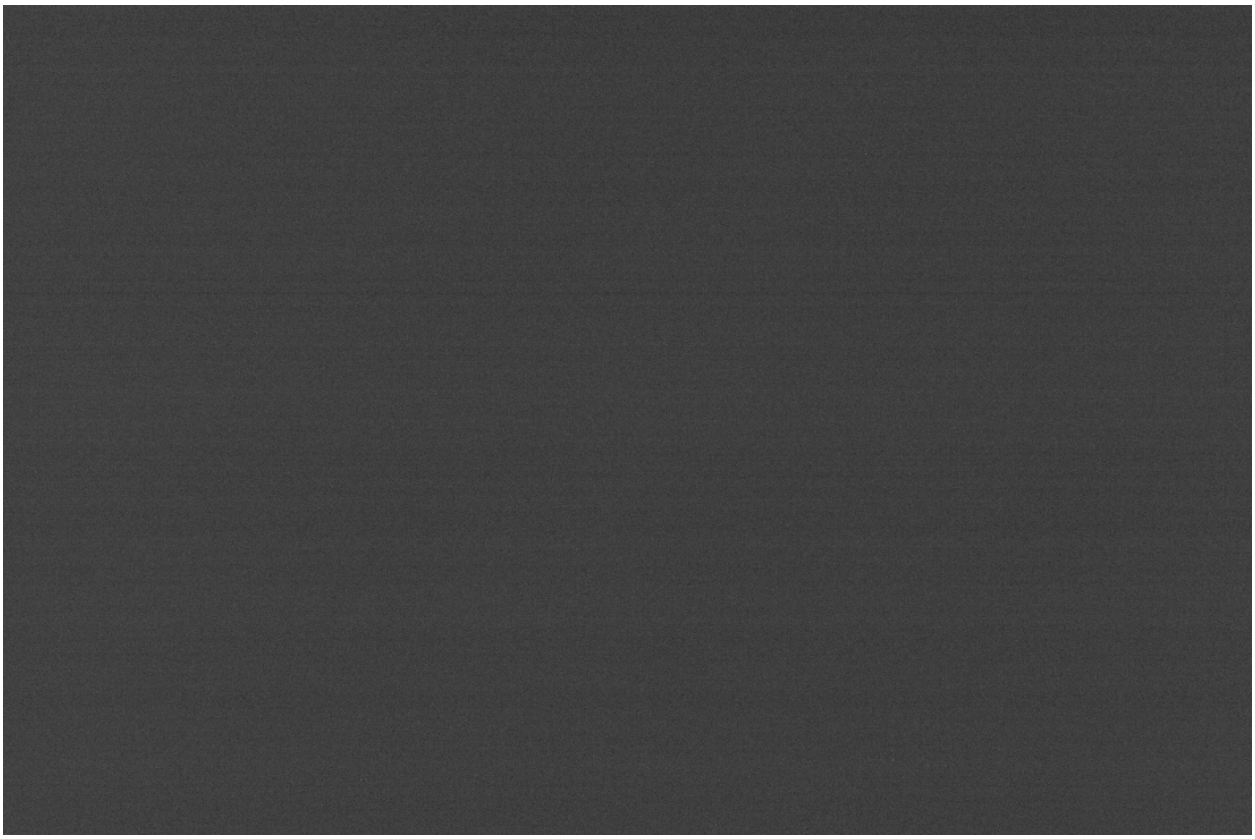


Abb. 7: Beispiel für ein Darkframe, aufgenommen mit einer Canon EOS 1100D mit einer Belichtungszeit von 300s und ISO 800.

Abb. 8: Eine Animation, die die Beseitigung des thermischen Signals dank der Darkframe-Subtraktion zeigt.

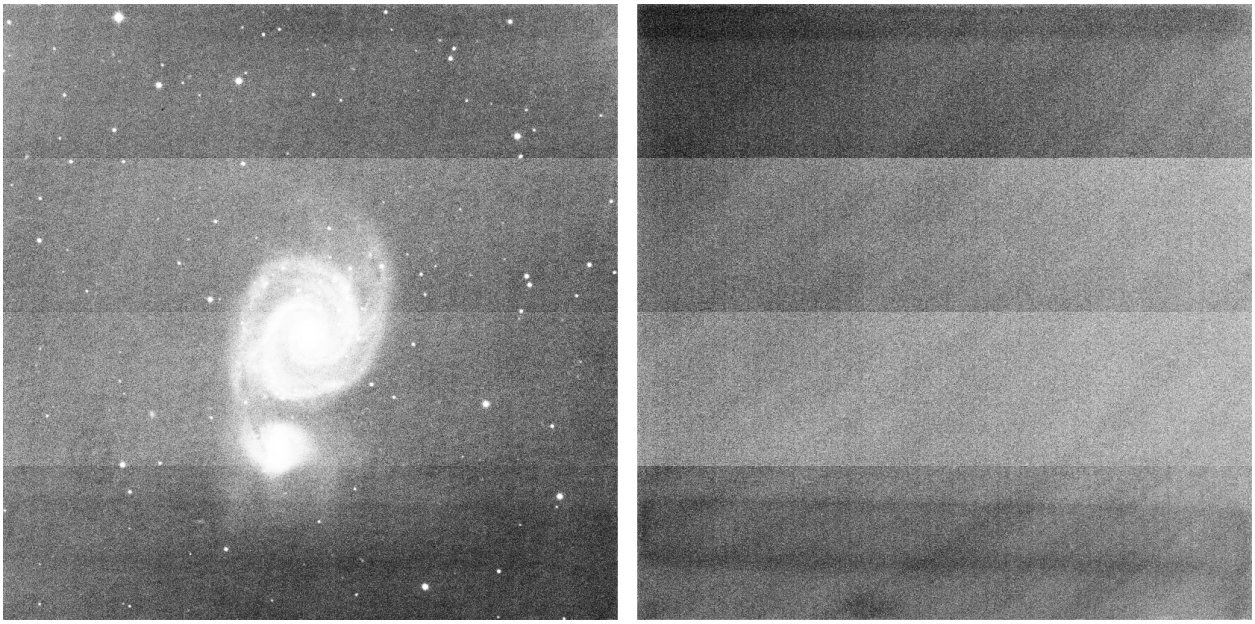


Abb. 9: Dies ist ein Light und das Master-Dark der **FLI ProLine 4240 Kamera**. Sie können 4 sehr charakteristische Bänder sehen, die durch die Vorverstärker verursacht werden, die sowohl bei den Lights als auch beim Master-Dark sichtbar sind. Die Bilder werden im Stretchmodus "Histogramm" angezeigt, um eventuelle Mängel hervorzuheben.

Wenn wir in diesem Fall jedoch den üblichen Arbeitsablauf verwenden, wird das Kalibrierungsergebnis sehr schlecht ausfallen. Dies liegt daran, dass das Master-Dark nicht unter den gleichen Belichtungsbedingungen aufgenommen wurde.

Die Lösung besteht daher darin, den Bias vom Dark zu subtrahieren, dann die Bias-Subtraktion mit denen der Bilder zu integrieren und das Kontrollkästchen für die Darkoptimierung zu aktivieren. Siril berechnet automatisch einen Koeffizienten, der auf die Darks angewendet wird. Hier wird **0,110** berechnet, was sehr kohärent ist, da es dem 10-fachen Unterschied zwischen Darks und Lights entspricht ($60/600 = 0,1$).

```
10:34:58: Preprocessing...
10:34:58: Normalisation value auto evaluated: 0.313
10:34:58: 13230 corrected pixels (0 + 13230)
10:34:59: Dark optimization of image 0: k0=0.110
10:34:59: Saving FITS: file pp_M51SDSSr_00002.fit, 1 layer(s), 2048x2048 pixels, 32 bits
```

Flats

Teleskope leuchten den Sensor im Allgemeinen nicht völlig gleichmäßig aus. Außerdem verursacht Staub auf den optischen Oberflächen und dem Sensor dunklere Muster im aufgenommenen Bild, und der Sensor selbst reagiert unterschiedlich auf die Anzahl der Photonen, die auf verschiedene Pixel auftreffen. Um diese Effekte zu korrigieren, muss jedes Lightframe durch das Masterflat geteilt werden, das der Median der Einzelbelichtungen eines homogenen und ungesättigten Bildausschnitts sein sollte.

Um ein Masterflat in Siril zu verwenden, klicken Sie auf die Schaltfläche rechts neben dem Texteintrag und durchsuchen Sie Ihre Dateien, um das richtige Master auszuwählen. Sie können sogar Masterflats aus einer Bibliothek verwenden, die in den *Einstellungen* definiert ist.

Abb. 10: Beim klassischen Workflow ist die Kalibrierung schlecht und Fehler werden nicht behoben. Das Bild wird Stretch-Modus "Histogramm" angezeigt, um eventuelle Mängel hervorzuheben.

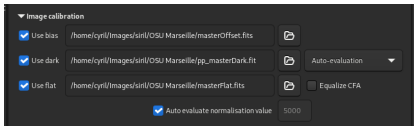


Abb. 11: Die Registerkarte „Kalibrierung“ sollte in einem solchen Fall ausgefüllt werden. Master-Flat und Master-Darks wurden durch Bias-Frames kalibriert.

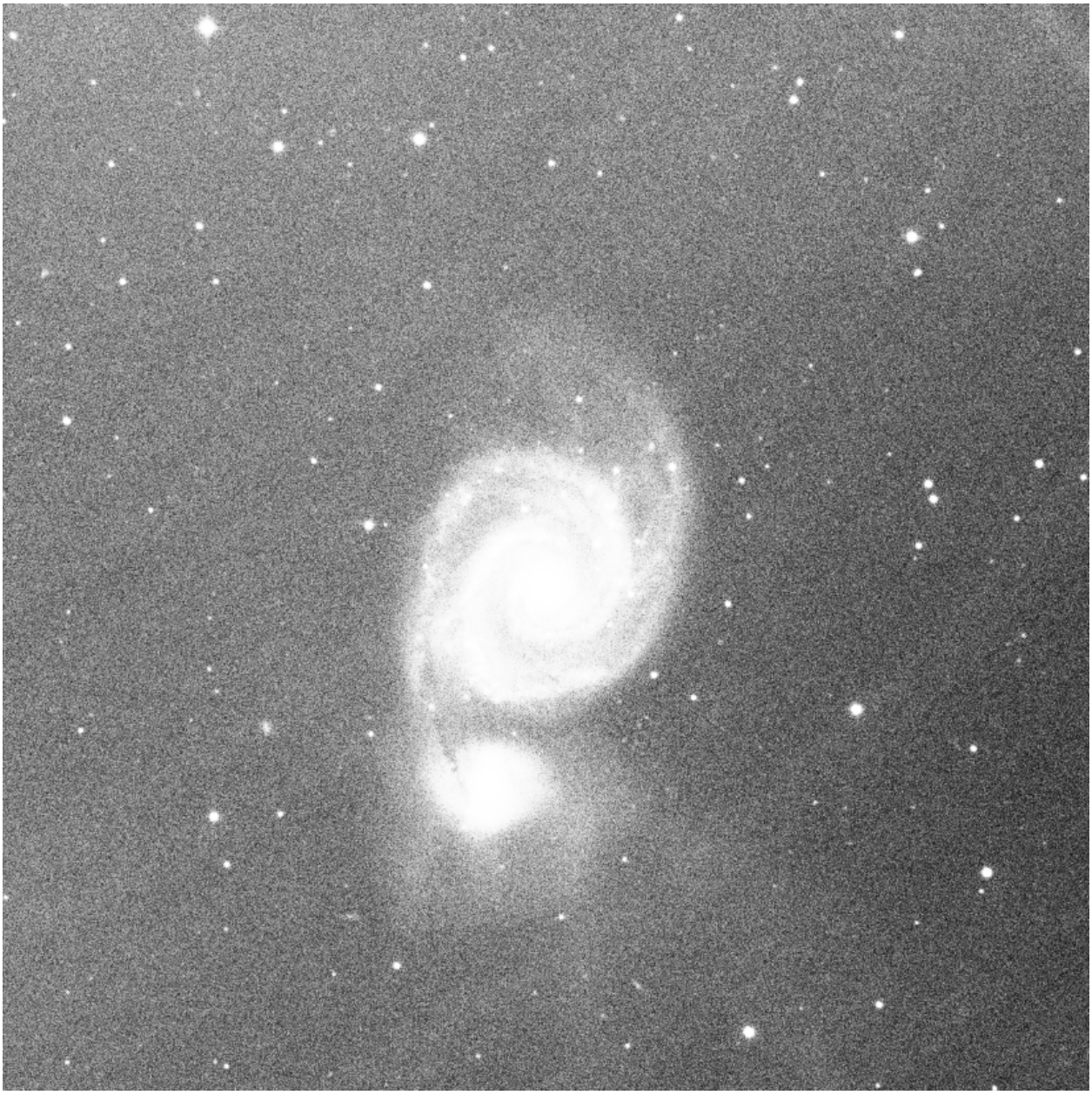


Abb. 12: Dank der Darkoptimierung ist die Kalibrierung korrekt. Der einzige sichtbare Rückstand ist etwas CCD-Fringing (Ausfransungen) im nahen IR-Bereich, das durch die Kalibrierung nicht entfernt werden kann. Das Bild wird im Stretchmodus "Histogramm" angezeigt, um eventuelle Mängel hervorzuheben.

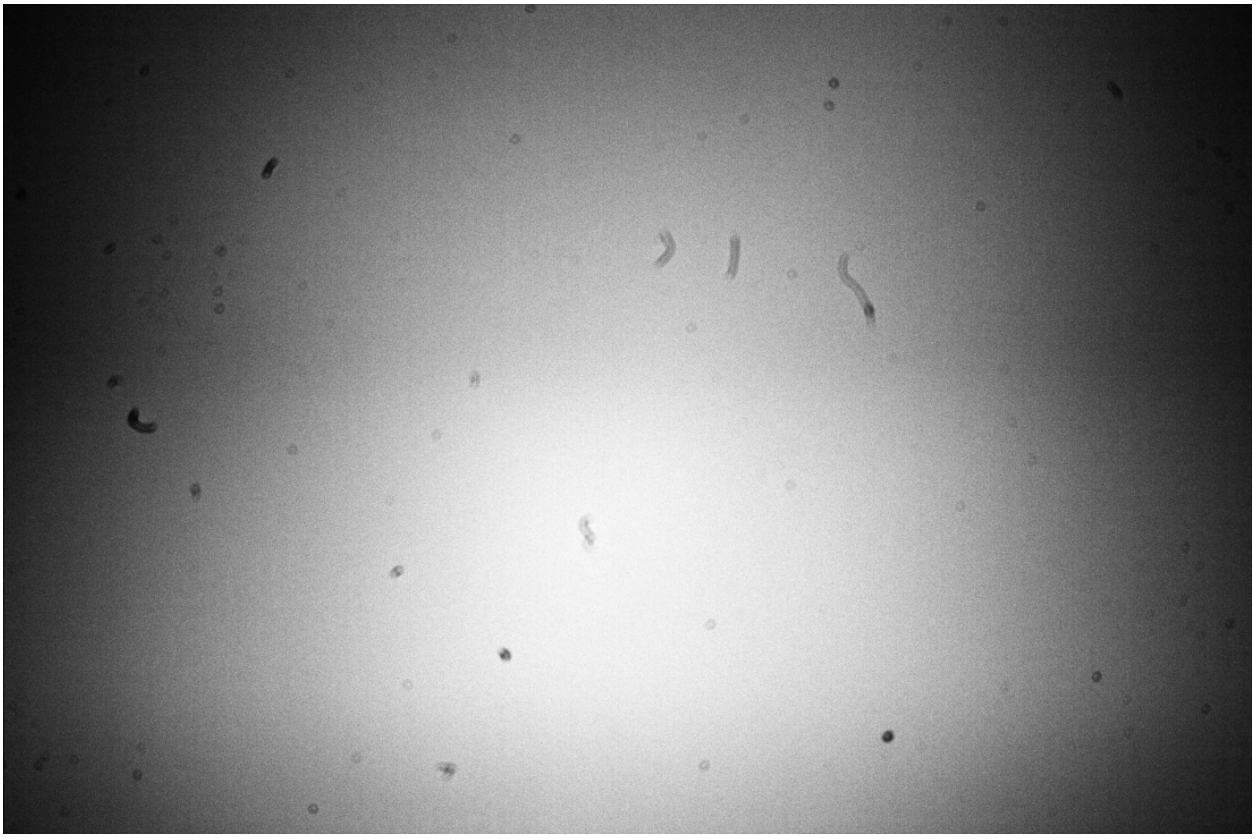


Abb. 13: Beispiel für ein Flatframe, das mit einer Canon EOS 1100D aufgenommen wurde. Der Staub, der sich auf dem optischen Pfad und insbesondere auf dem Sensor befindet, ist deutlich sichtbar. Die Vignettierung (Verdunkelung der Bildecken) ist ebenfalls deutlich sichtbar. Die Defekte werden durch den Anzeigemodus noch verstärkt. Außerdem wurde der Befehl *grey_flat* auf dieses Bild angewendet, um das Bayer-Muster zu beseitigen.

CFA-Ausgleich

Die Option *CFA ausgleichen* gleicht die mittlere Intensität der RGB-Ebenen in einem CFA-Flatframe aus. Dies entspricht der Anwendung des Befehls *grey_flat*.

Siril Kommandozeile

```
grey_flat
```

Gleicht die mittlere Intensität der RGB-Ebenen im geladenen CFA-Bild aus. Dies ist derselbe Vorgang, der bei der Kalibrierung bei Flats angewendet wird, wenn die Option „CFA ausgleichen“ verwendet wird

Automatische Ermittlung des Normalisierungswertes

Wenn die Option *Ermittle den Nomalisierungswert automatisch* aktiviert ist, wird Siril den Normalisierungswert automatisch ermitteln. Dieser Wert ist der Mittelwert des mit dem Masterbias kalibrierten Masterflats. Andernfalls wird der im Textfeld angegebene Wert berücksichtigt.

9.2.2 Kalibrierung der Lightframes

Die Kalibrierung der Lights besteht darin, Masterbias, -Dark und -Flat auf die astronomischen Bilder anzuwenden, um das unerwünschte Signal zu entfernen.

Warnung

Durch die Kalibrierung wird das Bildrauschen in keinem Fall verringert. Im Gegenteil, es vergrößert sich. Deshalb ist es wichtig, so viele Kalibrierungsbilder wie möglich aufzunehmen, z. B. Darks, um das Rauschen in den Mastern zu minimieren.

X-Trans AF Artefakt beheben

Die Option *X-Trans AF Artifakt korrigieren* hilft, die Fujifilm X-Trans Autofokus-Pixel zu korrigieren. Aufgrund des Phasendetektions-Autofokussystems erhalten die für den Autofokus verwendeten Fotozellen etwas weniger Licht als die umliegenden Fotozellen. Die Kamera gleicht dies aus und erhöht die Werte dieser speziellen Fotozellen, was zu einem sichtbaren Quadrat in der Mitte der Darks/Biasse führt. Diese Option hat keine Auswirkung auf das Bayer-Muster. Die Option ist nur aktiviert, wenn ein Master-Bias oder Master-Dark geladen und verwendet wird.

Kosmetische Korrektur

Die kosmetische Korrektur ist eine Technik, mit der fehlerhafte Pixel in Bildern korrigiert werden. Jeder Kamerasensor hat nämlich Sensorzellen, die nicht korrekt auf das Eintreffen von Photonen reagieren. Dies wird im Bild durch Pixel sichtbar, deren Werte sich stark von denen ihrer nächsten Nachbarn unterscheiden. Diese Pixel werden **Hotpixel** (Heiße Pixel) genannt, wenn der Wert viel höher ist, oder **Coldpixel** (Kalte Pixel), wenn er viel niedriger ist. Siril bietet zwei Algorithmen zur Korrektur dieser fehlerhaften Pixel, wenn die Option *Aktiviere Kosmische Korrektur* aktiviert ist.

Masterdark verwenden

Diese Methode erfordert das Vorhandensein eines Master-Darks. Siril sucht nach Pixeln, deren Abweichung vom Median das x-fache der Standardabweichung σ übersteigt. Dieser Wert ist sowohl für Hot- als auch für Coldpixel einstellbar.

Es ist möglich, die Anzahl der Pixel zu schätzen, die im kalibrierten Bild korrigiert werden, indem Sie auf die Schaltfläche *Abschätzen* drücken. Wenn der Wert der korrigierten Pixel in rot angezeigt wird, bedeutet dies, dass diese Zahl 1 % der Gesamtzahl der Pixel im Bild übersteigt. In diesem Fall sollten Sie den Wert des Koeffizienten erhöhen oder

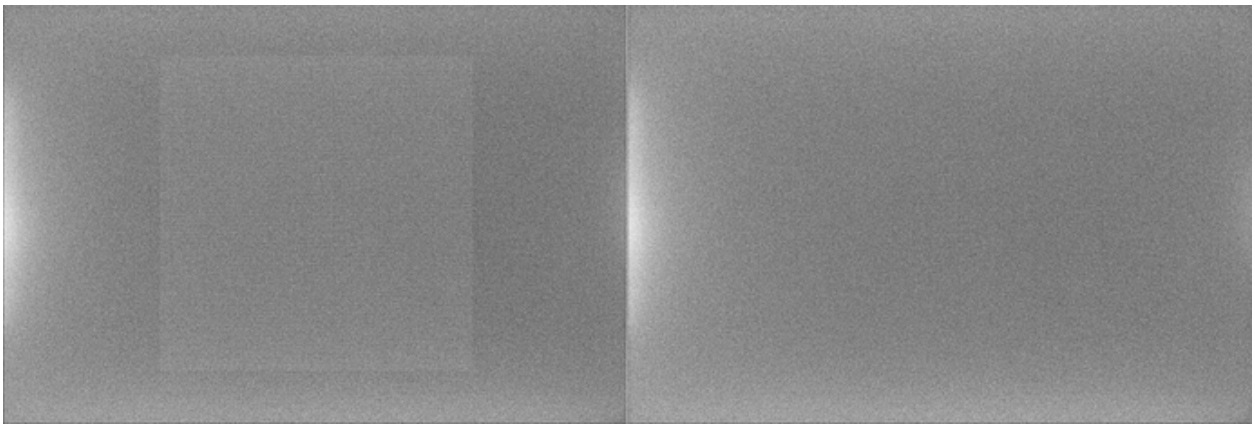
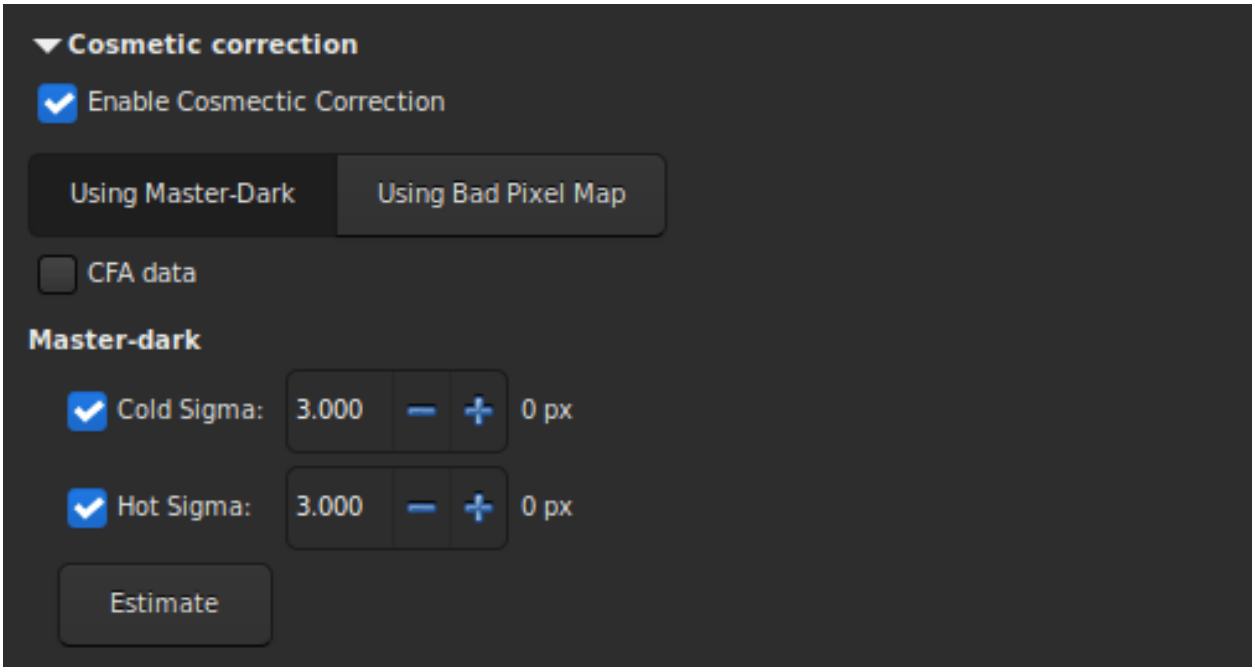


Abb. 14: Behebung des X-Trans-Artefakts durch den Algorithmus von Siril



die entsprechende Korrektur abwählen. Wenn die Bilder von einem Farbsensor stammen, ist es notwendig, die Option *CFA ausgleichen* zu aktivieren.

Eine Defekte Pixel Karte verwenden

Die andere Methode verwendet eine Datei, die die Koordinaten der defekten Pixel enthält. Diese Datei ist eine einfache Textdatei und kann zunächst mit dem Befehl *find_hot* erstellt werden. Die letzte Zeile wurde von Hand hinzugefügt und korrigiert eine beschädigte Spalte an Position $x = 1527$.

```
P 325 2855 H
P 825 2855 C
P 838 2855 H
P 2110 2855 H
P 2702 2855 H
P 424 2854 H
C 1527 0 H
```

Siril Kommandozeile

```
find_hot filename cold_sigma hot_sigma
```

Speichert eine Listendatei **filename** (Textformat) im Arbeitsverzeichnis, die die Koordinaten der Pixel enthält, die eine Intensität **hot_sigma** mal höher und **cold_sigma** mal niedriger als die vom geladenen Bild extrahierte Standardabweichung haben. Wir benutzten diesen Befehl generell bei einer Master-Dark-Datei. Der Befehl COSME kann diese Liste fehlerhafter Pixel auf ein geladenes Bild anwenden, siehe auch SEQCOSME, um sie auf eine Sequenz anzuwenden

Verweise: *cosme*, *seqcosme*

Die Zeile `P x y type` korrigiert das Pixel an den Koordinaten (x, y) type ist ein optionales Zeichen (C oder H), das Siril angibt, ob das aktuelle Pixel kalt oder heiß ist. Diese Zeile wird durch den Befehl FIND_HOT erzeugt, aber Sie können auch manuell Zeilen hinzufügen:

Die Zeile `C x 0 type` behebt die fehlerhafte Spalte an den Koordinaten x.

Die Zeile `L y 0 type` behebt die fehlerhafte Linie an den Koordinaten y.

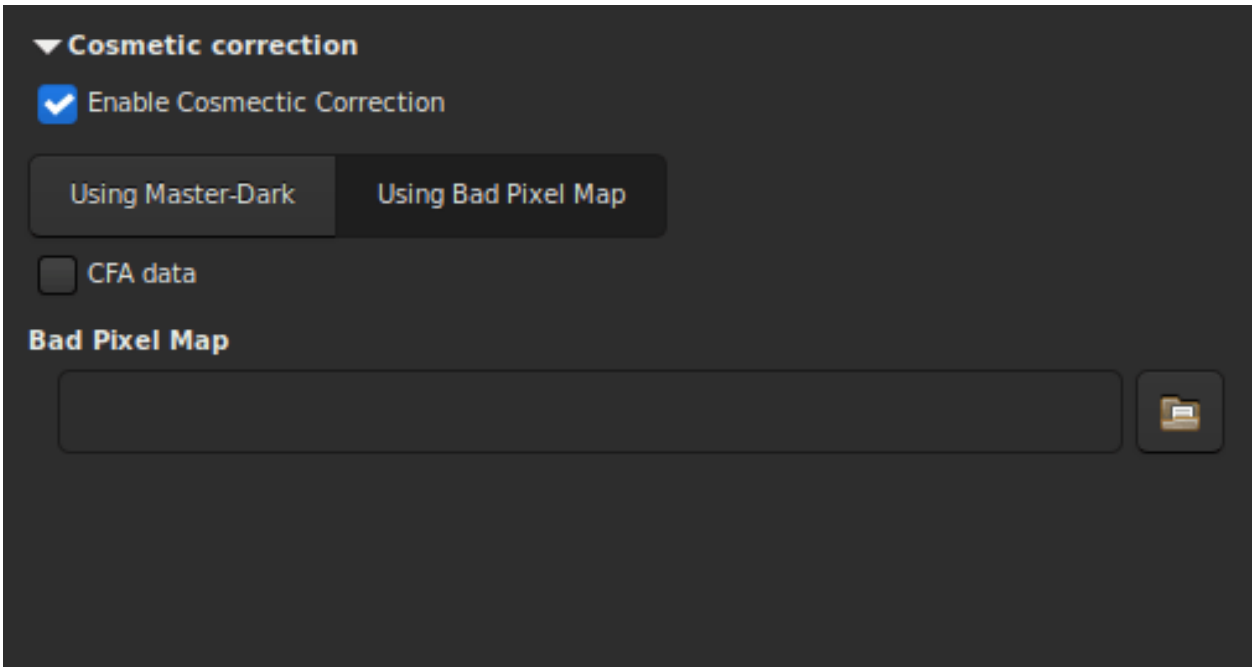
Diese Datei, die von Hand bearbeitet werden kann, ist als *Defekte Pixel* (Bad Pixel Map) zu laden.

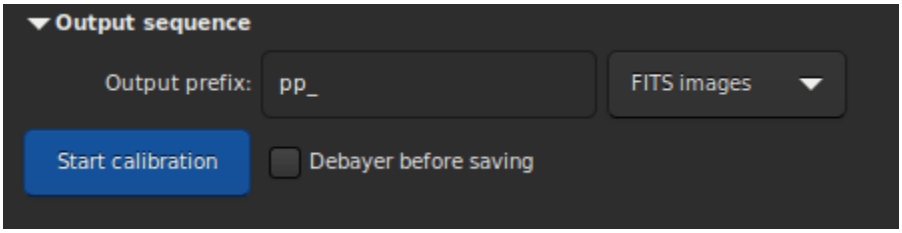
Wenn die Bilder von einem Farbsensor stammen, ist es notwendig, die Option *CFA ausgleichen* zu aktivieren.

Ausgabesequenz

In diesem Abschnitt sind die Optionen zusammengefasst, die auf die Ausgabe angewendet werden können.

- Das Eingabefeld *Ausgabepräfix* fügt den Ausgabebildern ein Präfix hinzu, um sie leichter identifizieren zu können. Standardmäßig lautet das Präfix `pp_`, was pre-processed (vorverarbeitet) bedeutet.
- In der Dropdown-Liste wird der Typ der Zielsequenz festgelegt.
  - FITS-Bilder: eine FITS-Datei pro Bild.
  - SER-Sequenz: eine SER-Datei für die gesamte Sequenz (begrenzt auf 16 Bit pro Kanal).


Cosmetic correction
Enable Cosmectic Correction
Using Master-Dark
Using Bad Pixel Map
CFA data
Bad Pixel Map


Output sequence
Output prefix:
pp_
FITS images
Start calibration
Debayer before saving

  - FITS Sequenz: eine FITS-Datei für die gesamte Sequenz.

- Letzte Option, *Vor dem Speichern die Bilder Debayern*. Aktivieren Sie diese Option, wenn Sie direkt nach der Kalibrierung einen Demosaicing-Algorithmus auf Ihre Bilder anwenden möchten. Auf diese Weise überspringen Sie einen manuellen Schritt, der einige Zeit in Anspruch nehmen kann.

Kommandozeilen

Siril Kommandozeile

```
calibrate sequencename [-bias=filename] [-dark=filename] [-flat=filename] [-cc=dark␣
↪[siglo sighi] || -cc=bpm bpmfile] [-cfa] [-debayer] [-fix_xtrans] [-equalize_cfa] [-
↪opt[=exp]] [-all] [-prefix=] [-fitseq]
```

Kalibrierung der Sequenz **Sequenzname** unter Verwendung der im Argument angegebenen bias-, dark- und flat-Dateien.

Für Bias kann anstelle eines Bildes ein einheitlicher Wert angegeben werden, indem ein mit einem =-Zeichen beginnender Anführungszeichenausdruck eingegeben wird, z. B. -bias="=256" oder -bias="=64*$OFFSET" (synthetischer Bias).

Standardmäßig ist die kosmetische Korrektur nicht aktiviert. Wenn Sie eine solche Korrektur wünschen, müssen Sie sie mit der Option **-cc=** angeben.

Sie können **-cc=dark** verwenden, um Hot- und Coldpixel aus dem Masterdark zu erkennen (ein Masterdark muss mit der Option **-dark=** angegeben werden), optional gefolgt von **siglo** und **sighi** für Cold- bzw. Hotpixel. Ein Wert von 0 deaktiviert die Korrektur. Wenn keine Sigmawerte angegeben werden, wird nur die Erkennung von Hotpixeln mit einem sigma von 3 angewendet.

Alternativ können Sie **-cc=bpm** gefolgt von dem Pfad zu Ihrer Bad Pixel Map verwenden, um anzugeben, welche Pixel korrigiert werden müssen. Eine Beispieldatei kann mit einem *find_hot*-Befehl auf einem Masterdark erhalten werden.

Drei Optionen gelten für Farbbilder (im CFA-Format): **-cfa** für kosmetische Korrekturen, **-debayer**, um Bilder vor dem Speichern zu entmosaikisieren, und **-equalize_cfa**, um die mittlere Intensität der RGB-Ebenen des Master-Flats auszugleichen, um eine Färbung des kalibrierten Bildes zu vermeiden.

Die Option **-fix_xtrans** widmet sich X-Trans-Dateien, indem sie eine Korrektur auf Dark- und Biasframes anwendet, um ein recheckiges Muster zu entfernen welches durch den Autofokus verursacht wird.

Es ist auch möglich, die Dunkelbildsubtraktion mit **-opt** zu optimieren, was die Bereitstellung von Bias und Dunkelmastern erfordert und automatisch den auf die Darks anzuwendenden Koeffizienten berechnet, oder den Koeffizienten dank des Belichtungs-Schlüsselworts **-opt berechnet =exp** berechnet.

Standardmäßig werden die als ausgeschlossen markierten Bilder nicht verarbeitet. Das Argument **-all** kann verwendet werden, um die Verarbeitung aller Bilder zu erzwingen, auch wenn sie als ausgeschlossen markiert sind.

Der Name der Ausgabesequenz beginnt mit dem Präfix "pp_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde.

Wenn **-fitseq** angegeben wird, ist die Ausgabesequenz eine FITS-Sequenz (einzelne Datei)

Siril Kommandozeile

```
calibrate_single imagename [-bias=filename] [-dark=filename] [-flat=filename] [-
↪cc=dark [siglo sighi] || -cc=bpm bpmfile] [-cfa] [-debayer] [-fix_xtrans] [-
↪equalize_cfa] [-opt[=exp]] [-prefix=]
```

Kalibriert das Bild **imagename** unter Verwendung der im Argument angegebenen Bias-, Dark- und Flat-Dateien.

Für Bias kann anstelle eines Bildes ein einheitlicher Wert angegeben werden, indem ein mit einem =-Zeichen beginnender Anführungszeichenausdruck eingegeben wird, z. B. -bias="=256" oder -bias="=64*$OFFSET" (synthetischer Bias).

Standardmäßig ist die kosmetische Korrektur nicht aktiviert. Wenn Sie eine solche Korrektur wünschen, müssen Sie sie mit der Option **-cc=** angeben.
Sie können **-cc=dark** verwenden, um Hot- und Coldpixel aus dem Masterdark zu erkennen (ein Masterdark muss mit der Option **-dark=** angegeben werden), optional gefolgt von **siglo** und **sighi** für Cold- bzw. Hotpixel. Ein Wert von 0 deaktiviert die Korrektur. Wenn keine Sigmawerte angegeben werden, wird nur die Erkennung von Hotpixeln mit einem sigma von 3 angewendet.
Alternativ können Sie **-cc=bpm** gefolgt von dem Pfad zu Ihrer Bad Pixel Map verwenden, um anzugeben, welche Pixel korrigiert werden müssen. Eine Beispieldatei kann mit einem *find_hot*-Befehl auf einem Masterdark erhalten werden.

Drei Optionen gelten für Farbbilder (im CFA-Format): **-cfa** für kosmetische Korrekturen, **-debayer**, um Bilder vor dem Speichern zu entmosaikisieren, und **-equalize_cfa**, um die mittlere Intensität der RGB-Ebenen des Master-Flats auszugleichen, um eine Färbung des kalibrierten Bildes zu vermeiden.
Die Option **-fix_xtrans** widmet sich X-Trans-Dateien, indem sie eine Korrektur auf Dark- und Biasframes anwendet, um ein recheckiges Muster zu entfernen welches durch den Autofokus verursacht wird.
Es ist auch möglich, die Dunkelbildsubtraktion mit **-opt** zu optimieren, was die Bereitstellung von Bias und Dunkelmastern erfordert und automatisch den auf die Darks anzuwendenden Koeffizienten berechnet, oder den Koeffizienten dank des Belichtungs-Schlüsselworts **-opt berechnet =exp** berechnet
Der Ausgabedateiname beginnt mit dem Präfix "pp_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

9.2.3 Verstehen, wie die Flats die Lights korrigieren

In diesem Abschnitt geht es darum, einen Einblick zu geben welche Rolle die verschiedenen Komponenten bei der Korrektur der Lights durch die Flats spielen.

Wir lassen hier alle Überlegungen zum Rauschen außer Acht (auch hier verschwindet das Rauschen nicht durch Subtraktion oder Division, sondern durch Mittelwertbildung über viele Realisierungen desselben Zufallsprozesses). Wir lassen auch besondere räumliche Muster wie Ampglow oder Staub außer Acht.

Wenn wir versuchen, die Intensität der Hintergrundpixel in den verschiedenen Frames, die wir haben, zu quantifizieren,

können wir die folgenden Ausdrücke schreiben:

$$L = a - b \times \left(x - \frac{W}{2}\right)^2 + d_{\text{rate}} \times t_{\text{lights}} + o \tag{9.1}$$

$$D = d_{\text{rate}} \times t_{\text{lights}} + o \tag{9.2}$$

$$F = K\left(a - b \times \left(x - \frac{W}{2}\right)^2\right) + o \tag{9.3}$$

$$O = o \tag{9.4}$$

wobei, L für Lights, D für Darks, F für Flats und O für Bias steht.

Für die Lights ist L der erste Teil eine räumliche Beleuchtungskomponente, *d.h.*, $a - b(x - \frac{W}{2})^2$. Wir haben hier eine quadratische Variation mit einem Maximalwert a in der Mitte des Frames der Breite W gewählt, also etwa in der Mitte des Sensors. Dies ist nicht die exakte räumliche Form der Vignettierung, aber es ist eine ausreichend gute Annäherung, um zu verstehen, wie sie funktioniert. Zusätzlich zu diesem räumlichen Beleuchtungsterm gibt es einen Term, der mit der Belichtungszeit variiert und gewöhnlich als Dunkelstrom bezeichnet wird ($d_{\text{rate}} \times t_{\text{lights}}$), der aber nicht von der Position des Pixels auf dem Sensor abhängt. Und schließlich gibt es einen Sockelwert, *d.h.* den Offset. Dieser Versatz ist in jedem aufgenommenen Bild vorhanden, so dass wir ihn in allen Ausdrücken finden.

Die Darks D werden nicht belichtet, sie tragen nur den Dunkelstromterm, mit der gleichen Intensität wie die Lichter, da sie mit der gleichen Belichtungszeit aufgenommen werden sowie dem gleichen Offset.

Die Flats F haben auch einen räumlichen Term, der proportional zu dem Term der Lights ist. Der Faktor K, der größer als 1 ist, zeigt einfach, dass ihre Intensität größer ist. Um dies zu schreiben, müssen wir nur annehmen, dass die Pixel linear auf die Anzahl der Photonen reagieren, die sie sammeln, was sinnvoll ist. Wir hätten auch einen Dunkelstromterm schreiben können, der proportional zur Belichtungszeit der Flats ist. Aber wenn diese Zeit nicht signifikant ist, können wir davon ausgehen, dass sie vernachlässigbar ist. Wenn das nicht der Fall ist, bedeutet das, dass man Darkflats aufnehmen oder zumindest deren Höhe abschätzen muss.

Und schließlich messen die Offsets O nur den Offset-Pegel.

Zur Veranschaulichung dieser Komponenten haben wir diese Ausdrücke als Kurven in Bezug auf die Position auf ein Frame gezeichnet, und wir ermutigen Sie, dasselbe zu tun und mit den Eingaben zu spielen.

- $a = 200$[ADU]
- $b = 0.0003$[ADU/px^2]
- $d_{\text{rate}} = 1$[ADU/s]
- $t_{\text{lights}} = 10$[s]
- $o = 2048$[ADU]
- $W = 1000$[px]

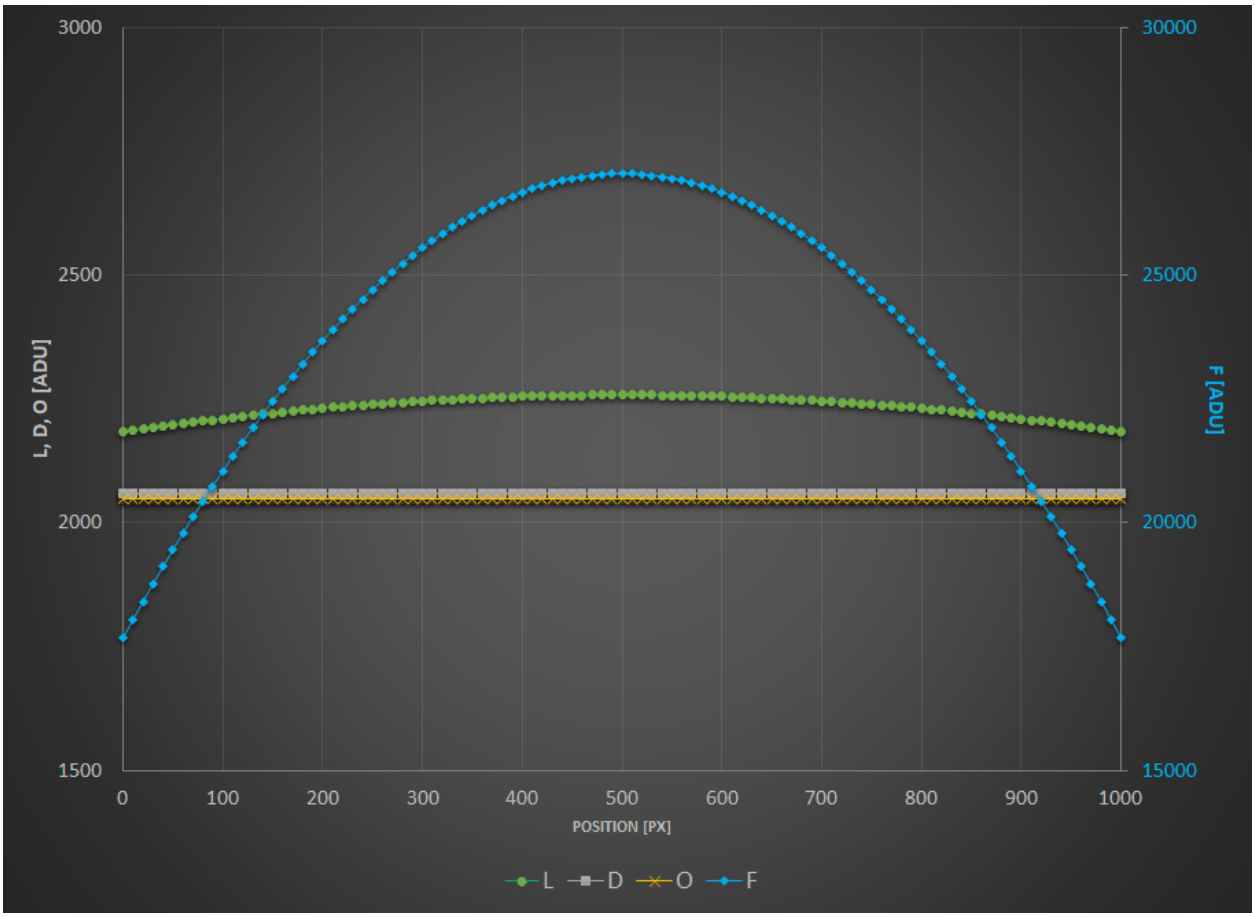
Die Werte L, D und O in ADU werden auf der linken Skala angegeben, während F auf der rechten Skala angegeben wird.

Was bedeutet es nun, die Lights zu kalibrieren? Wenn Sie Ihre Lights kalibrieren, führen Sie den folgenden Vorgang durch:

$$L_c = \frac{L - D}{F - O}.$$

Der Begriff $F - O$ ist ein Flat, von dem Sie die Offset-Ebene abgezogen haben (unabhängig davon, ob es sich um eine Masterbias oder einfach einen festen Wert handelt). Dies ist der Vorgang, der vor dem Stapeln des Masterflats durchgeführt wird. Und der Ausdruck $L - D$ steht für ein Light, von dem Sie den Dunkelstrompegel und den Offset subtrahiert haben, *d.h.* ein Masterdark. Ersetzt man die oben gezeigten Ausdrücke, erhält man folgendes Ergebnis:

$$L_c = \frac{1}{K}.$$

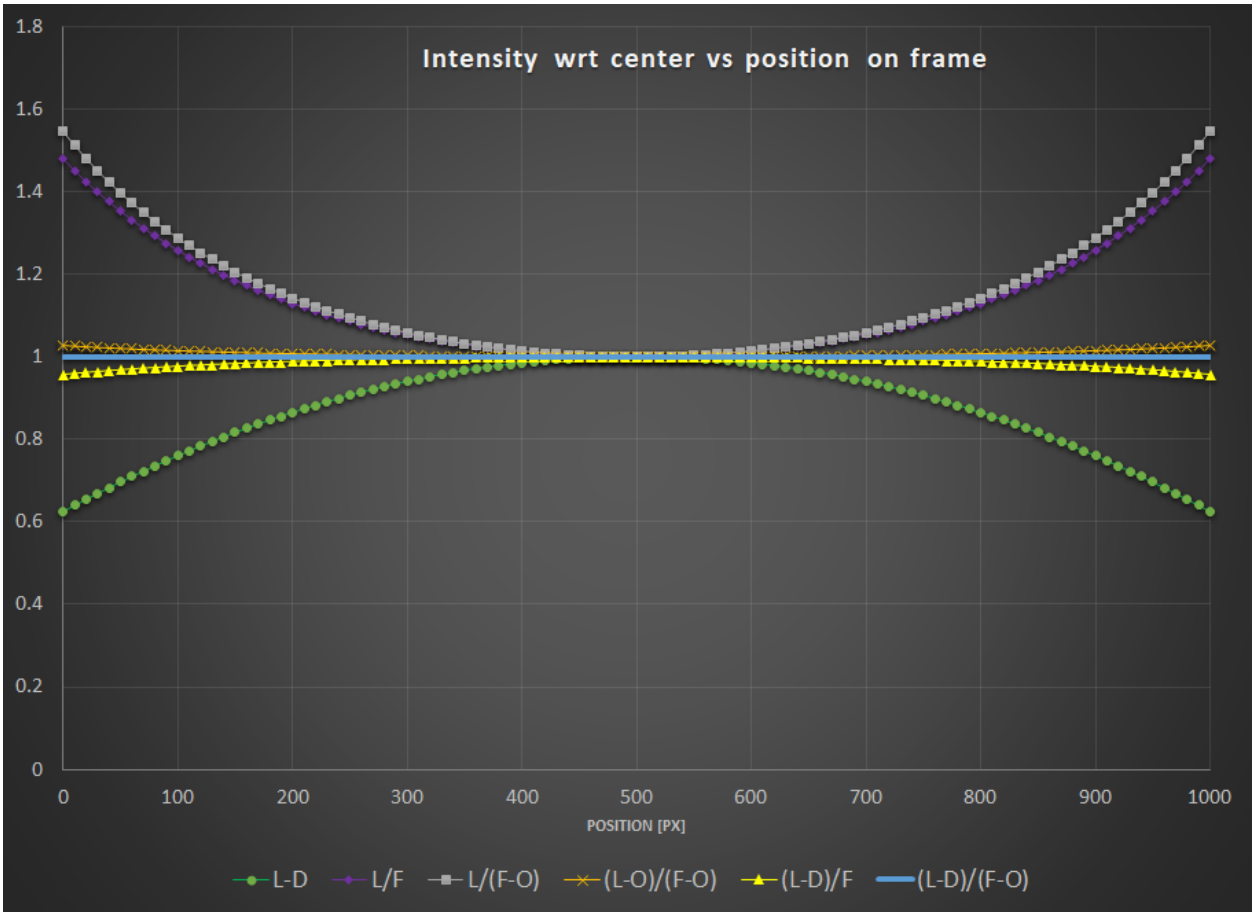

3000
2500
2000
1500
30000
25000
20000
15000
L, D, O [ADU]
F [ADU]
0
100
200
300
400
500
600
700
800
900
1000
POSITION [PX]
L
D
O
F

Es bleibt kein räumlicher Variationsterm übrig, Sie haben Ihre bereits mit Flats korrigiert! Siril ermittelt einen sinnvollen Wert in ADU (und nicht $1/K$), wenn Sie *Ermittle den Normalisierungswert automatisch* in der Registerkarte *Kalibrierung* aktivieren.

Und Sie können es mit jeder anderen Kombination versuchen, keine andere wird die räumlichen Abweichungen beseitigen.

Zur Veranschaulichung haben wir unten die Ergebnisse verschiedener Kombinationen aufgetragen. Um alles auf die gleiche Skalierung zu bringen, sind alle Ergebnisse auf die gleiche Intensität von 1 in der Mitte des Bildes normalisiert. Die folgenden Tests werden dargestellt:

- $L - D$: Sie haben nur Darks aufgenommen.
- L/F : Sie haben nur Flats erstellt.
- $L/(F - O)$: Sie haben Flats aufgenommen und diese durch einen Offset (entweder ein Masterbias oder einen synthetischen Bias) korrigiert.
- $(L - O)/(F - O)$: Sie haben nur Flats um den Offset korrigiert. Aber Sie haben auch den Offset von Ihren Lights abgezogen.
- $(L - D)/F$: Sie haben Flats und Darks aufgenommen, aber keinen Offset/Bias.
- $(L - D)/(F - O)$: Sie haben Alles lehrbuchmäßig durchgeführt.



Interessanterweise können Sie feststellen:

- $L - D$ zeigt offensichtlich keine Korrektur der Vignettierung.
- Aber sowohl L/F als auch $L/(F - O)$ zeigen eine Überkorrektur oder umgekehrte Vignettierung.

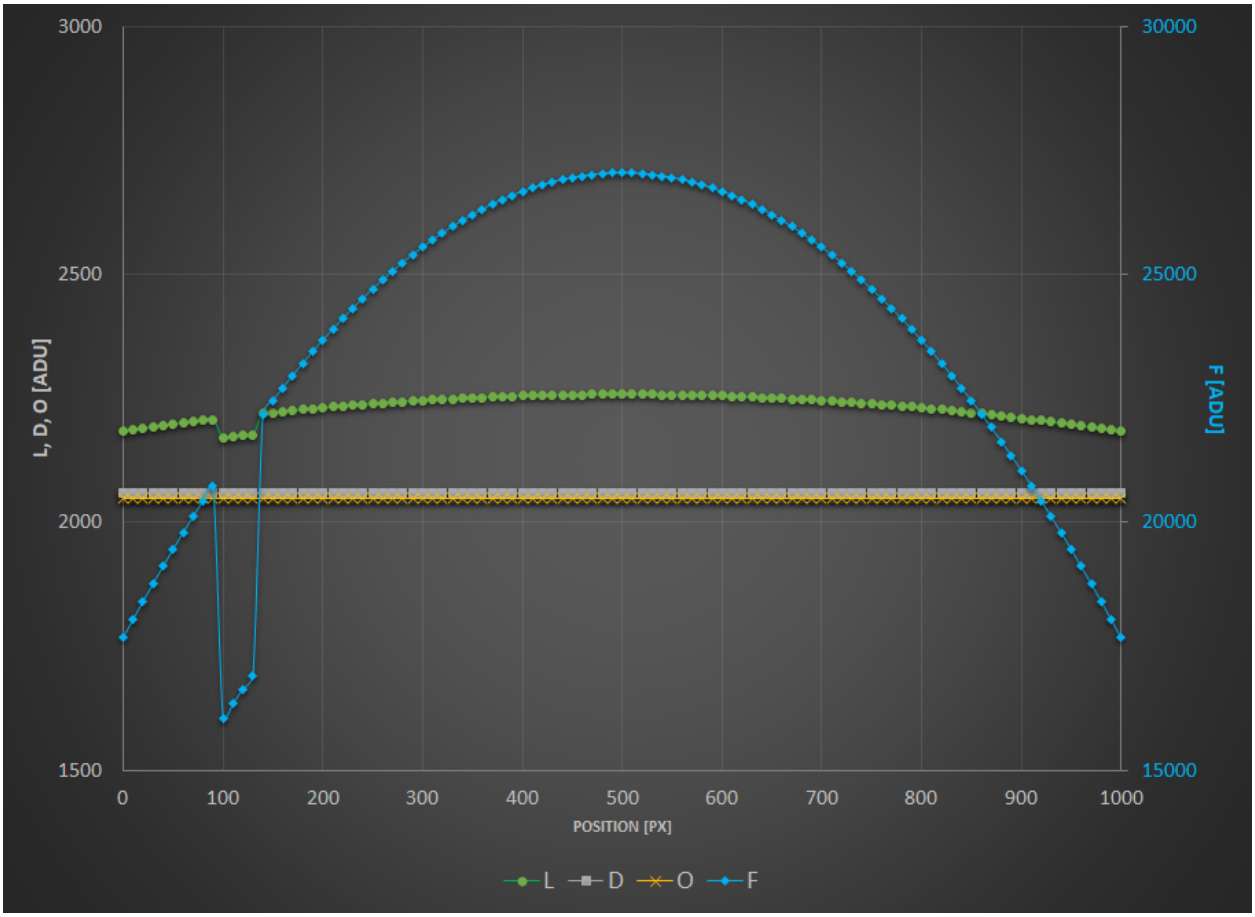
- Um dem optimalen Ergebnis sehr nahe zu kommen, zeigen $(L-D)/F$ und $(L-O)/(F-O)$ ein nahezu flaches Feld. Dies hängt natürlich davon ab, wie viel Dunkelstrom Ihr Sensor hat und wie viel Vignettierung Ihr Optical Train (Zusammenstellung aller Komponenten vor der Kamera) aufweist.
- Die Referenzkalibrierung ergibt ein flaches Feld.

Die Schlussfolgerungen, die Sie aus dem oben Gesagten ziehen können, sind:

- Es ist besser, wenn Sie Ihre Lights mit einem Offset (Masterbias oder einfach einem festen Wert) korrigieren, wenn Sie keine Darks gemacht haben.
- Noch besser: Wenn Sie keine Zeit haben, eine Reihe von Darks zu machen, lohnt es sich wahrscheinlich, mindestens ein Dark zu machen, dessen Median zu messen und dieses (synthetische) Darklevel von Ihren Lights abzuziehen. Dadurch wird natürlich weder das Ampglow korrigiert noch die Hot-Pixel-Korrektur ermöglicht, aber Ihre Lights werden zumindest flach sein!

Und was ist mit Staub...?

Damit Ihre Flats auch diese unschönen Flecken korrigieren können, müssen Sie leider auch alle Kalibrierungsframes in die Gleichung einbeziehen. Wir haben ein kleines lokales ADU-Defizit in den Lights und Flats hinzugefügt, um diesen Effekt zu veranschaulichen.



Wie Sie sehen können, kann nur die Kombination $(L-D)/(F-O)$ sie korrigieren.

Zur weiteren Veranschaulichung der obigen Gleichungen und Kurven eignet sich nichts besser als ein Beispiel aus der Praxis. Alle Bilder unten wurden mit freundlicher Genehmigung von G. Attard zur Verfügung gestellt.

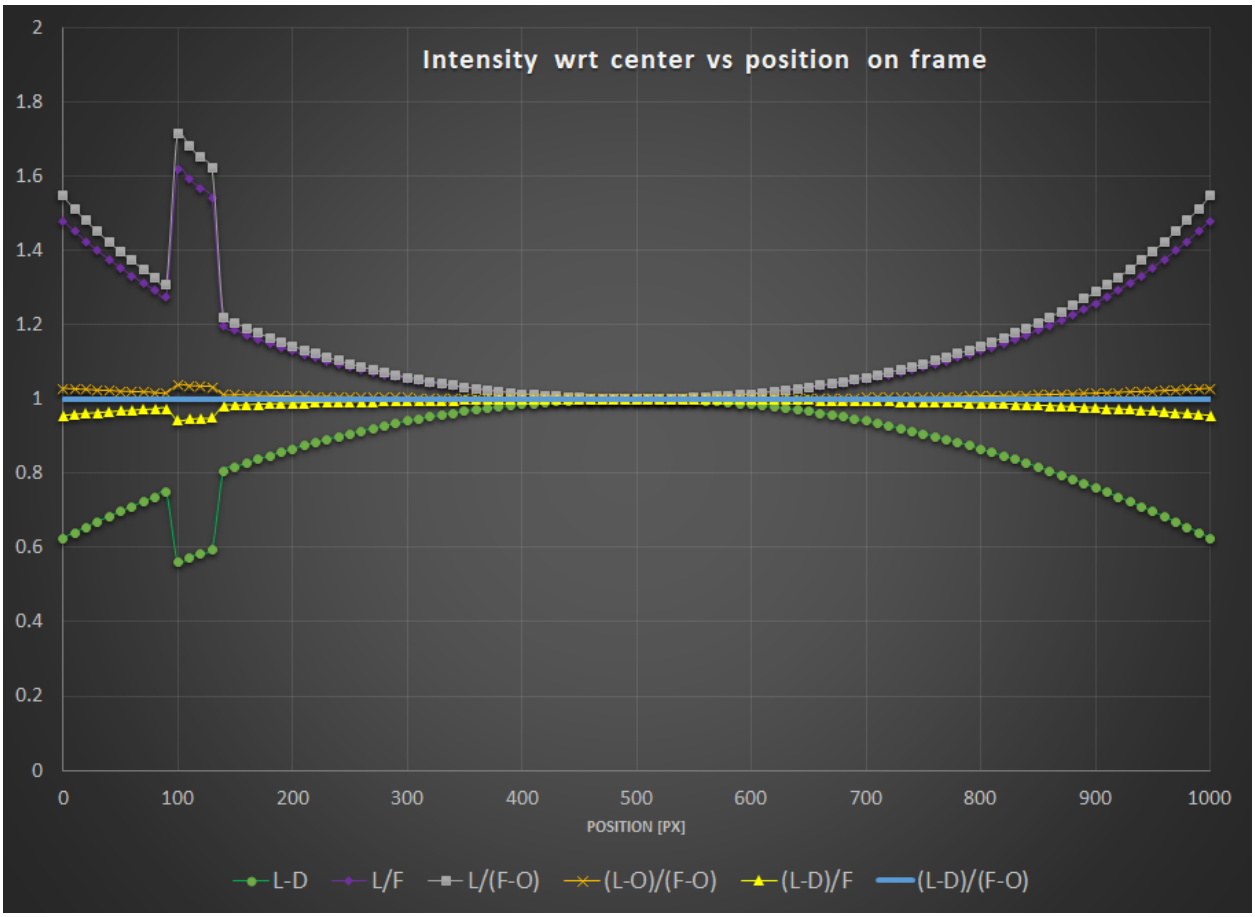

Intensity wrt center vs position on frame
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000
POSITION [PX]
L-D
L/F
L/(F-O)
(L-O)/(F-O)
(L-D)/F
(L-D)/(F-O)

Abb. 15: $L - D$

Abb. 16: L/F

Abb. 17: $L/(F - O)$

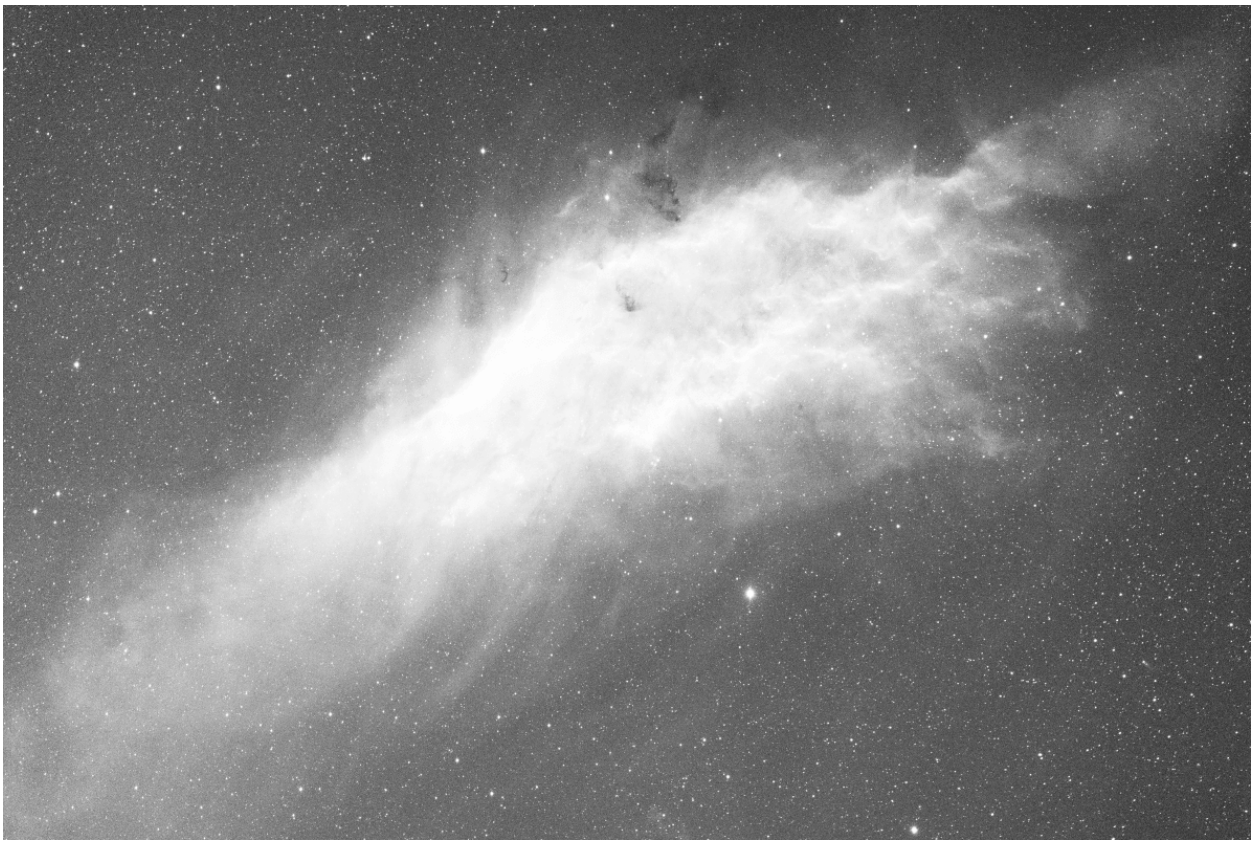


Abb. 18: $(L - O)/(F - O)$

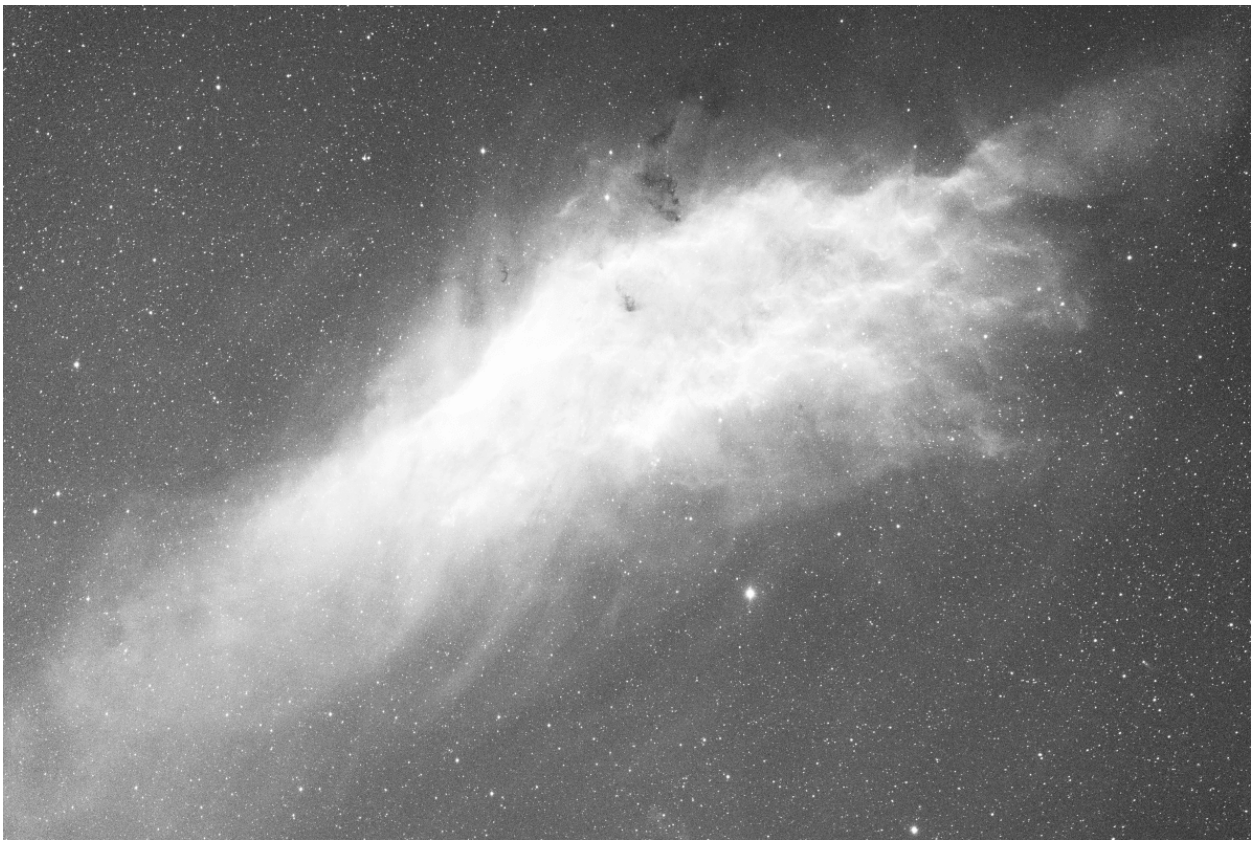


Abb. 19: $(L - D)/(F - O)$

9.2.4 Fehlerbehebung bei der Kalibrierung

Die Kalibrierung ist rechnerisch ein sehr einfacher Schritt und kann nicht fehlschlagen, wenn die Eingabedaten den Erwartungen für astronomische Bilder entsprechen.

Allerdings werden Benutzer regelmäßig mit Situationen konfrontiert, in denen die kalibrierten Bilder nicht korrekt sind. In diesem Abschnitt geben wir Ihnen einen Überblick über die möglicherweise auftretenden Probleme und wie Sie diese vermeiden können.

Zunächst einmal ist das *Statistiktool* eine unschätzbare Hilfe zum Verständnis von Problemen und wird in den meisten Fällen zur Behebung von Problemen verwendet.

- Bei der Analyse der Statistiken eines Master-Darks muss dieses zunächst schwarz sein. Dies liegt daran, dass diese Bilder bei verschlossener Kamera aufgenommen wurden und es keinen Grund gibt, warum eine der Fotozellen privilegiert sein sollte. Das Bild muss so aussehen, als wäre es mit einem Monochromsensor aufgenommen worden, wobei die Bayer-Matrix nicht sichtbar ist. Unten sehen Sie ein Beispiel, bei dem der Master-Dark einen unerwünschten Farbabgleich für diesen Bildtyp erfahren hat. Dadurch ist es nicht mehr schwarz und das Bayer-Muster ist sichtbar. Ein solches Darkframe ist unbrauchbar und muss neu erstellt werden.
- Während der Nachtsitzung ist es sehr wichtig, den OFFSET-Wert für alle Bilder auf den gleichen Wert einzustellen. Insbesondere ist es zwingend erforderlich, für die Paare Darks/Lights und Bias/Flats die gleiche Einstellung zu haben. Die Nichterfüllung der ersten Bedingung kann zum Verlust erheblicher Daten führen (Pixel werden auf der linken Seite des Histogramms abgeschnitten). Wenn beide Bedingungen nicht erfüllt sind, wird höchstwahrscheinlich verhindert, dass Ihre Bilder korrekt mit den Flats korrigiert werden (siehe *Abschnitt oben*).
- Überprüfen Sie die Helligkeits- und Dunkelwerte: Der Mittelwert der Lightframes muss ausreichend höher sein als der der Darkframes, um zu vermeiden, dass Bilder voller Pixel mit negativen ADU-Werten entstehen.
- Wenn Sie die gleichen Einstellungen für Darks und Bias verwendet haben, sollten deren Mittelwerte sehr nahe beieinander liegen (zumindest bei einer gekühlten Kamera). Andernfalls kann es sein, dass Sie ein Lichtleck haben, das sich auf Ihre Darkframes ausgewirkt hat (Biasframes sind weniger empfindlich, da sie mit einer viel kürzeren Belichtungszeit aufgenommen werden). Überprüfen Sie daher immer Ihren Master-Dark, um festzustellen, ob sich in der Mitte ein Farbverlauf oder ein hellerer Fleck befindet. Dies ist nicht mit Ampglow zu verwechseln, der bei bestimmten Kameras normal ist.
- Wir empfehlen dringend, dass Sie Ihre Bilder auf die gleiche Weise aufnehmen: gleiche Software / gleicher Computer oder Astrobox / gleiches Bildformat. Tatsächlich kann es sein, dass jedes Programm seine eigenen Schreibkonventionen verwendet und Bilder möglicherweise nicht mehr miteinander kompatibel sind. Wir hören oft von Benutzern, die alle ihre Bilder mit einer Astrobox machen und die Flats am nächsten Tag direkt mit ihrer DSLR erstellen. In diesem Fall sind die Bilder oft unterschiedlich groß, sodass eine Kalibrierung nicht möglich ist.
- Ein beim Ausführen eines Skripts häufig auftretender Fehler ist das Vorhandensein von JPG-Bildern in einem der Eingabeordner (Darks/Biases/Flats/Lights), meist Schnappschüsse, die von der Erfassungssoftware zum schnelleren Durchsuchen gespeichert werden. Die Folge eines solchen Fehlers ist, dass die Kalibrierung fehlschlägt und stoppt und sich darüber beschwert, dass die Bilder nicht die gleiche Größe haben. Da JPG-Bilder bereits debayert sind, verfügen sie tatsächlich über drei Kanäle, während RAW-Bilder nur einen haben. Entfernen Sie alle JPG-Bilder aus den Eingabeordnern, um dieses Problem zu beheben.
- Stellen Sie sicher, dass die Flats nicht überbelichtet sind. Flatframes werden verwendet, um die Empfindlichkeitsschwankungen des Sensors von Pixel zu Pixel zu korrigieren. Wenn einige Pixel überbelichtet sind, wird ihre tatsächliche Lichtempfindlichkeit möglicherweise nicht genau dargestellt, was zu falschen Korrekturen während des Kalibrierungsprozesses führt. Eine überbelichtetes Flatframe ist die Garantie für eine fehlerhafte Kalibrierung.

  Um nach überbelichteten Pixeln zu suchen, können Sie ein Flatframe laden und *Bildverarbeitung → Histogrammtransformation...* verwenden, um das Histogramm des Bildes anzuzeigen. Der Schnappschuss unten zeigt, dass eine der Berge rechts abgeschnitten ist. Als Vorsichtsmaßnahme sollten Sie stets darauf achten, dass der rechte Ausläufer des Peaks, der am weitesten rechts liegt, nicht über 80 % liegt, um zu vermeiden, dass Sie in einen Bereich geraten, in dem Ihr Sensor möglicherweise nichtlinear wird.

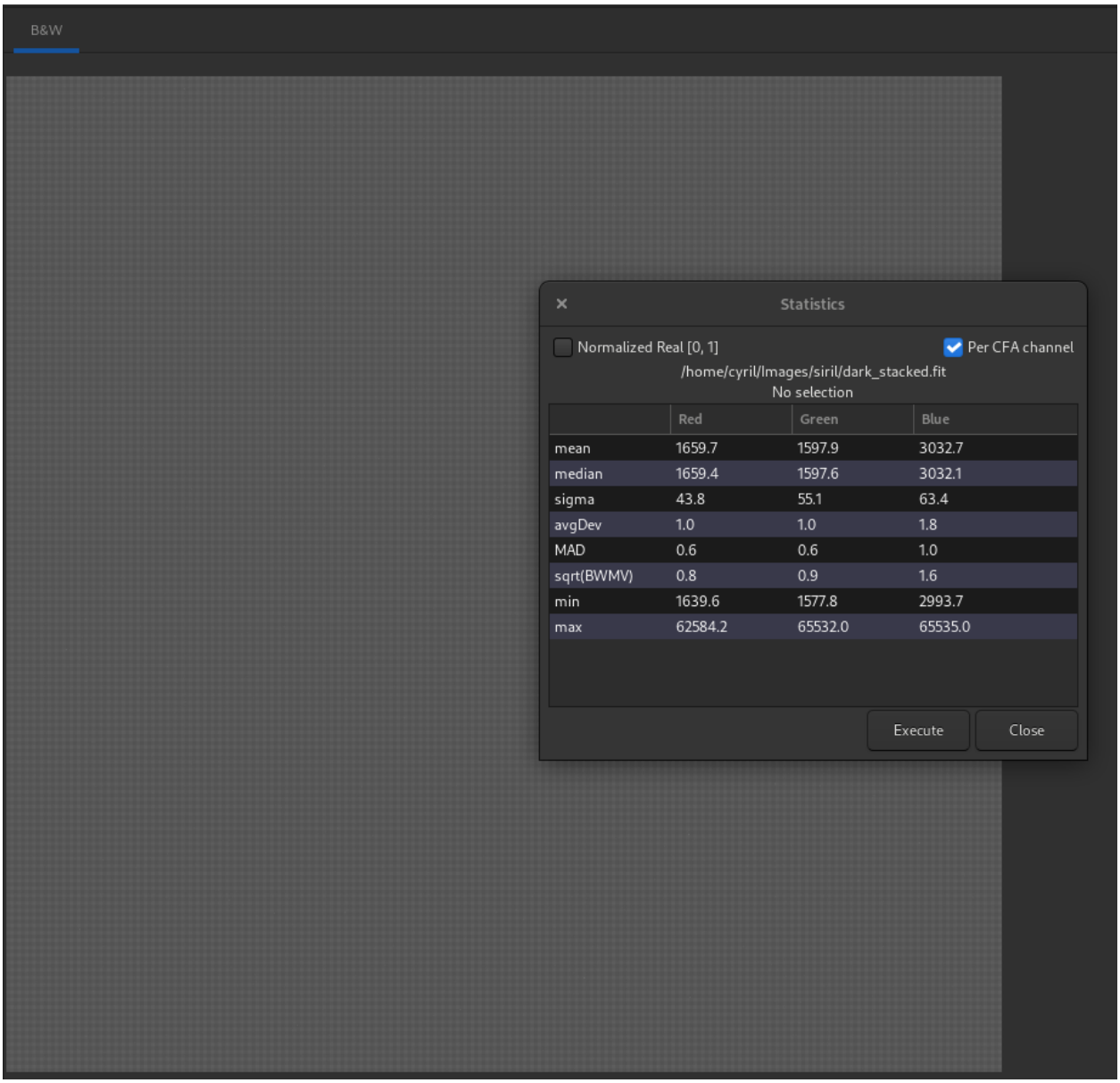


Abb. 20: Ein genauer Blick auf die Statistiken zeigt, dass der Medianwert jedes Kanals unterschiedlich ist, obwohl sie identisch (oder nahezu identisch) sein sollten. Auch die Bayermatrix ist deutlich erkennbar.

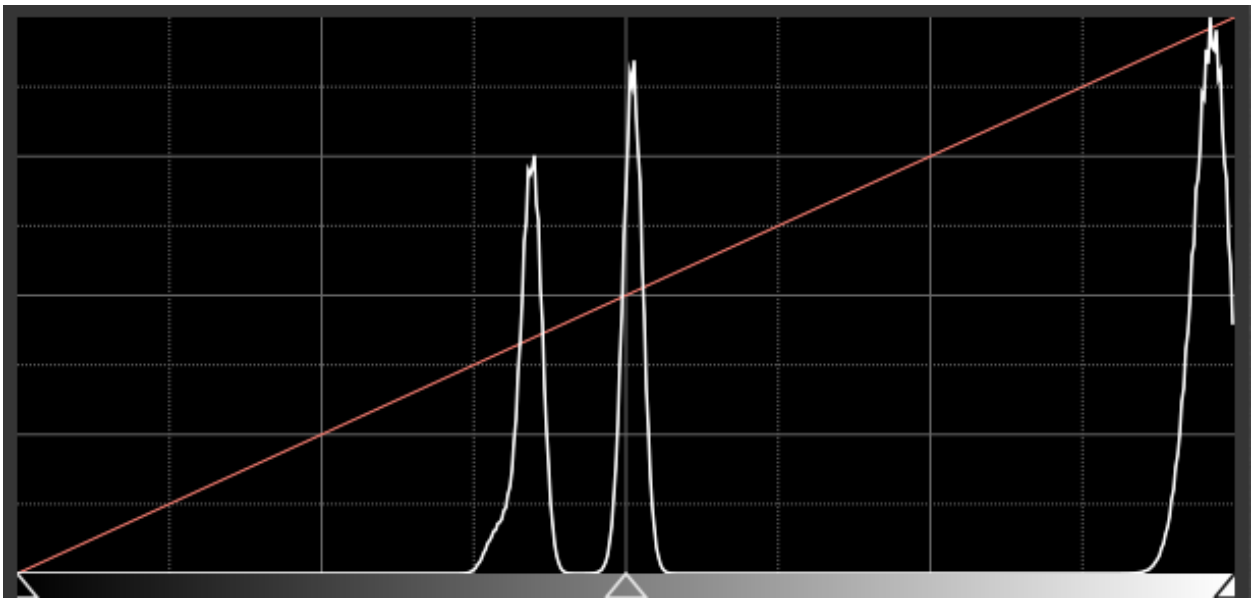


Abb. 21: Weiß-Clipping eines Flatframes. Wenn dies geschieht, bedeutet das, dass Sie den Gain oder die Belichtungszeit verringern sollten.

9.3 Registrierung

Die Registrierung ist im Grunde der Prozess der Ausrichtung der Bilder einer Sequenz, um sie anschließend verarbeiten zu können. Alle im Folgenden beschriebenen Verfahren berechnen die Transformation, die auf jedes Bild anzuwenden ist, um es am Referenzbild der Sequenz auszurichten.

Die Stärke von Siril liegt in der großen Vielfalt der angebotenen Ausrichtungsalgorithmen. Jede Methode wird im Folgenden erläutert. Durch Drücken der Schaltfläche *Führe Registrierung aus* wird die Registrierung der Sequenz gestartet.

Es ist möglich, die **Registrierungsebene** zu wählen. Grün ist die Standardeinstellung für Farbbilder, Luminanz für Monochrombilder. Das Zeichen (*) hinter dem Namen des Kanals bedeutet, dass für diese Ebene bereits Registrierungsdaten verfügbar sind. Bei der Verarbeitung von Bildern werden die Registrierungsdaten der Standardebene entnommen, wenn sie verfügbar sind (bei RGB-Bildern: Grün, sonst Blau und dann Rot).

9.3.1 Theorie

Verfahren der Registrierung

Was wir Registrierung nennen, ist in Wirklichkeit ein dreistufiger Prozess:

1. Erkennen der abzugleichenden Merkmale in allen Bildern
2. Berechnen einer lineare Funktion zwischen einem Bild und dem Referenzbild
3. Anwendung der berechneten Transformation auf jedes Bild, um neue Bilder zu erhalten

Je nach gewählter Registrierungsmethode werden die 3 Schritte zu einem einzigen Prozess zusammengefasst (oder nicht). Siril verwendet je nach gewählter Registrierungsmethode die sinnvollste Standardeinstellungen (ob die berechnete Transformation angewendet werden soll oder nicht), aber ein Verständnis der internen Mechanismen kann Ihnen helfen, dieses Verhalten Ihren Bedürfnissen entsprechend zu ändern.

Algorithmen

In der nachstehenden Tabelle sind die verschiedenen Algorithmen aufgeführt, die für die ersten beiden Schritte (Erkennung und Transformationsberechnung) verwendet werden.

Registrierungs-methode	Merkmalserkennung	Transformationsberechnung	New seq
Global	*Dynamisches PSF*	Dreiecksvergleich und RANSAC	Y
Global in 2 Durchgängen			N
1-2-3 Sterne	PSF *Minimierung* in der Auswahlbox	Singulärwert-Zerlegung (2-3 Sterne) Differenz (1 Stern)	N
Bildmusterausrichtung	Kreuzkorrelation der Auswahlbox		N
KOMBAT	Max der Faltung im räumlichen Bereich auf dem Auswahlfeld		N
Komet	PSF *Minimierung* in der Auswahlbox	Verschiebungen des Geschwindigkeitsvektors durch Zeitstempel	Y(*)
Manuell	Deine Augen	Deine Hand	N
Astrometrie	*Dynamisches PSF*	von der astrometrischen Lösung	N

Bemerkung

(*) Die Kometenregistrierung erstellt eine neue Sequenzdatei (standardmäßig mit dem Präfix *comet_*) und erstellt „neue“ Bilder. Wenn in der Eingabesequenz Registrierungsdaten vorhanden waren, werden die Kometenverschiebungen mit diesen bereits vorhandenen Registrierungsdaten erstellt. Die erstellten „neuen“ Bilder sind lediglich symbolische Links zu den Eingabebildern, sodass sie keinen zusätzlichen Speicherplatz belegen. Diese in Version 1.4 eingeführte Änderung wurde vorgenommen, um zu verdeutlichen, ob eine Sequenz mit Registrierungsdaten nur eine Sternausrichtung oder eine Stern- und Kometenverschiebungsausrichtung hatte. Beim Exportieren einer neuen Sequenz ist nun deutlich, dass zusätzliche Verschiebungen enthalten sind. Außerdem können Sie dieselbe Eingabesequenz verwenden und mehrere Sequenzen erstellen, die auf verschiedene sich bewegende Objekte ausgerichtet sind.

In der nachfolgenden Tabelle sind die mit den einzelnen Methoden kompatiblen Transformationen aufgeführt, und es wird angegeben, ob auch eine Entzerrung verfügbar ist.

Registrierungsmethode	Verschiebung	Euklidisch	Ähnlichkeit	Affin	Homographie	Entzerren
Global	Subpixel		x	x	x	x
2-Durchgänge	Subpixel		x	x	x	x
1-2-3 Sterne	Subpixel (1-2-3)	(2-3)				
Bildmusterausrichtung	Pixel					
KOMBAT	Pixel					
Komet	Subpixel					
Manuell	Pixel					
Astrometrie					x	x

Es ist auch wichtig zu wissen, wie die registrierte Sequenz in den Stackingprozess eingespeist wird, der in der Regel direkt nach der Registrierung erfolgt:

- Wenn die Transformation nur aus pixelweisen Verschiebungen besteht, kann der Stackingalgorithmus diese Verschiebungen beim Einlesen der Bilder sofort verwenden. Dies bedeutet, dass Sie keine "registrierten Bilder" erzeugen müssen. Das spart Speicherplatz und macht die Interpolation überflüssig. Dies geht natürlich auf Kosten

einer weniger genauen Registrierung (d. h. ohne Subpixel-Genauigkeit), wird aber im Allgemeinen bei Planeten-/Lucky-Imaging verwendet, bei denen die Abtastrate gering ist. Dies kann auch mit einer Registrierungsmethode angewendet werden, die Subpixel-Verschiebungen berechnet. Während des Stacking-Prozesses werden die Verschiebungen auf Pixelgenauigkeit gerundet. In allen anderen Fällen, d. h. wenn dem Stacking eine Sequenz zugeführt wird, bei der die Registrierung komplexere Transformationen als nur Verschiebungen berechnet hat, die registrierten Bilder aber nicht gespeichert wurden, gibt Siril eine Warnung aus und fordert Sie auf, die registrierten Bilder zu exportieren, bevor Sie mit dem Stacking fortfahren.

- In allen anderen Fällen müssen die transformierten Bilder nach der Berechnung der Transformationen gespeichert werden, bevor mit dem Stacking fortgefahren werden kann, im Allgemeinen mit dem Präfix `r_`.

Bildtransformationen

Lineare Transformationen

Siril verwendet lineare Transformationen mit unterschiedlichen Freiheitsgraden, um ein Bild dem Referenzbild zuzuordnen:

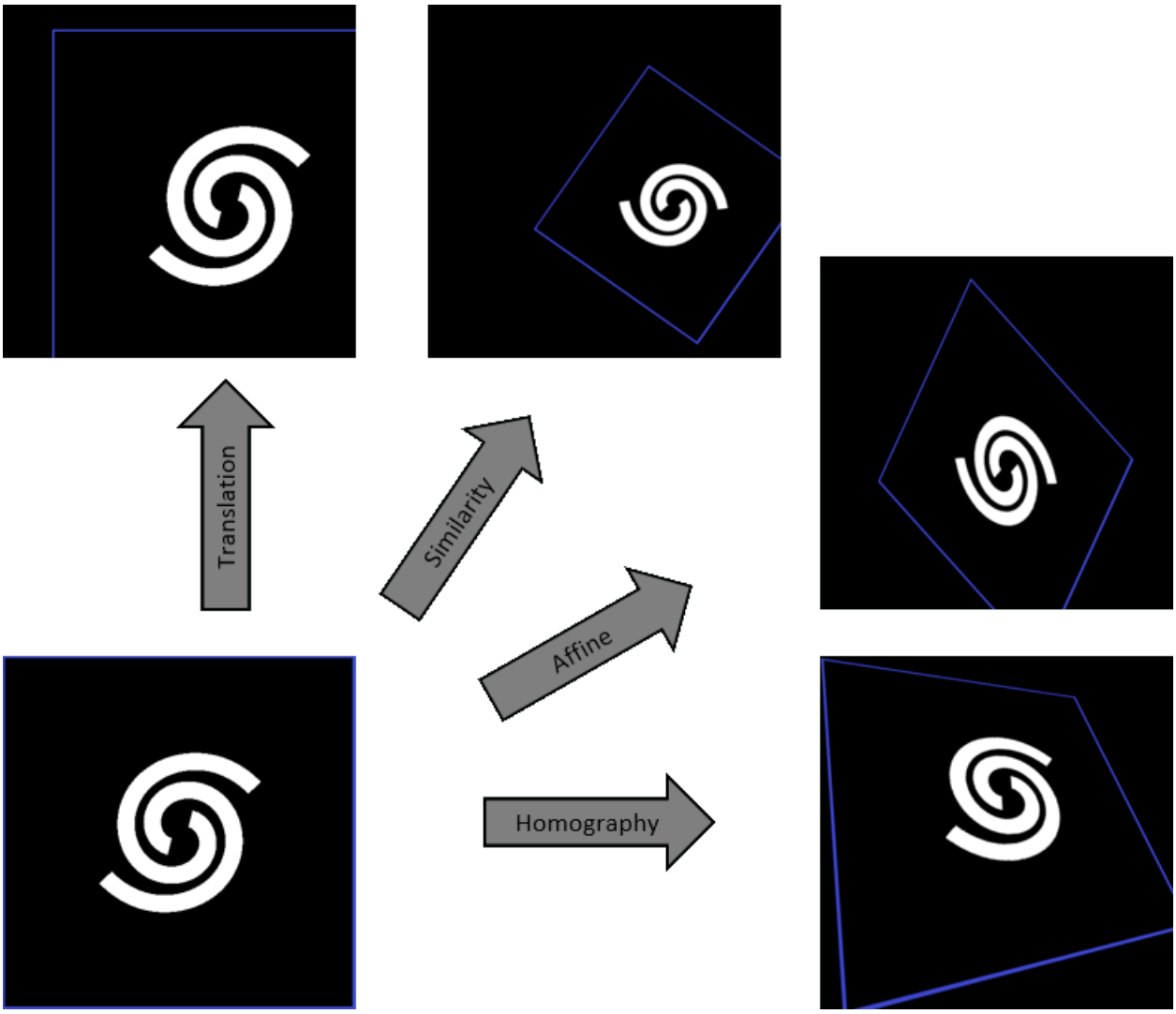
- **Verschiebung** ist eine Abbildung mit 2 Freiheitsgraden (x/y-Verschiebungen), die sich gut für Bilder ohne Verzerrung, Skalierung und Felddrehung eignet. Zur Definition der Transformation muss nur 1 Sternpaar aufeinander registriert werden.
- **Euklidisch** ist eine starre Abbildung mit 3 Freiheitsgraden (x/y-Verschiebungen + eine Drehung), für Bilder ohne Verzerrung und Skalierung. Zur Definition der Transformation müssen mindestens 2 Sternpaare aufeinander ausgerichtet werden.
- **Ähnlichkeit** ist eine Abbildung mit 4 Freiheitsgraden (eine Skalierung, eine Drehung und x/y-Verschiebungen), die starrer als die Homografie ist und sich gut für Bilder ohne Verzerrungen eignet. Es müssen mindestens 2 Paare von Sternen übereinstimmen, um die Transformation zu definieren.
- **Affine** ist eine Abbildung mit 6 Freiheitsgraden (zwei Skalierungen, eine Scherung, eine Drehung und x/y-Verschiebungen), die starrer ist als die Homografie und sich gut für Bilder mit geringer Verzerrung eignet. Für die Definition der Transformation müssen mindestens 3 Sternpaare aufeinander registriert werden.
- **Homographie** ist die Standardtransformation, die eine Transformation mit 8 Freiheitsgraden verwendet, um die Bilder auf den Referenzrahmen zu verzerren. Dieses Verfahren ist im Allgmeinen gut geeignet und wird für Weitwinkelaufnahmen dringend empfohlen. Um die Transformation zu definieren, müssen mindestens 4 Sternpaare aufeinander registriert werden.

Verzerrungen

Seit Version 1.3 kann Siril auch Verzerrungen berücksichtigen, und zwar für einige der Registrierungsmethoden, die in dieser *Tabelle* aufgeführt sind. Die von Siril verarbeiteten Verzerrungskoeffizienten folgen der SIP-Konvention. Diese Konvention geht davon aus, dass die Pixelkoordinaten korrigiert werden müssen, BEVOR versucht wird, sie durch eine lineare Transformation abzubilden. Im WCS-Jargon wird dies als *vorherige* Verzerrung bezeichnet (im Gegensatz zu einer *sequentiellen* Verzerrung).

Diese Koeffizienten werden während des Registrierungsprozesses zweimal verwendet:

- Die unter "verzerrten" Bedingungen erkannten Sternpositionen werden zunächst korrigiert, sowohl im auszurichtenden Bild als auch im Referenzbild (oder den projizierten Sternpositionen für die astrometrische Registrierung). Anschließend wird die *lineare Transformation* berechnet, die die Sterne vom aktuellen Bild auf das Referenzbild abbildet.
- Beim Exportieren des registrierten Bildes wird zunächst die Verzerrung korrigiert und dann linear projiziert, um es an das Referenzbild anzupassen. Beachten Sie, dass dies tatsächlich in einem einzigen Vorgang geschieht (die Pixelzuordnung wird als Zusammensetzung dieser nichtlinearen Korrektur und dann der linearen Projektion berechnet), um eine zweimalige Interpolation der Pixelwerte zu vermeiden. Das Referenzbild wird dieser Korrektur ebenfalls ohne die lineare Vorprojektion unterzogen.


Translation
Similarity
Affine
Homography

Referenzbild

Dies ist das Bild, das als gemeinsame Referenz für die Berechnung der Transformationen verwendet wird, die alle Bilder der Sequenz auf dieses spezielle Bild transformieren.

Falls es nicht manuell eingestellt wird, wird das Referenzbild nach den folgenden Kriterien ausgewählt:

- wenn die Sequenz bereits registriert wurde, ist es das beste Bild, d. h. das mit der niedrigsten FWHM oder der höchsten Qualität, je nach Art der Registrierung
- Andernfalls ist es das erste Bild der Sequenz, das nicht ausgeschlossen wird.

Um ein Bild als Referenz anzugeben, können Sie dies tun:

- Öffnen Sie den *Bilderliste*, wählen Sie das Bild, das als neue Referenz festgelegt werden soll, und klicken Sie auf die Schaltfläche *Referenz-Bild*.
- Verwenden Sie den Befehl *setref*. Zum Beispiel, wenn Sie Bild #10 als Referenz festlegen möchten:

```
setref 10
```

Siril Kommandozeile

```
setref sequencename image_number
```

Legt das Referenzbild der im ersten Argument angegebenen Sequenz fest. **image_number** ist die fortlaufende Nummer des Bildes in der Sequenz, nicht die Nummer im Dateinamen, beginnend bei 1

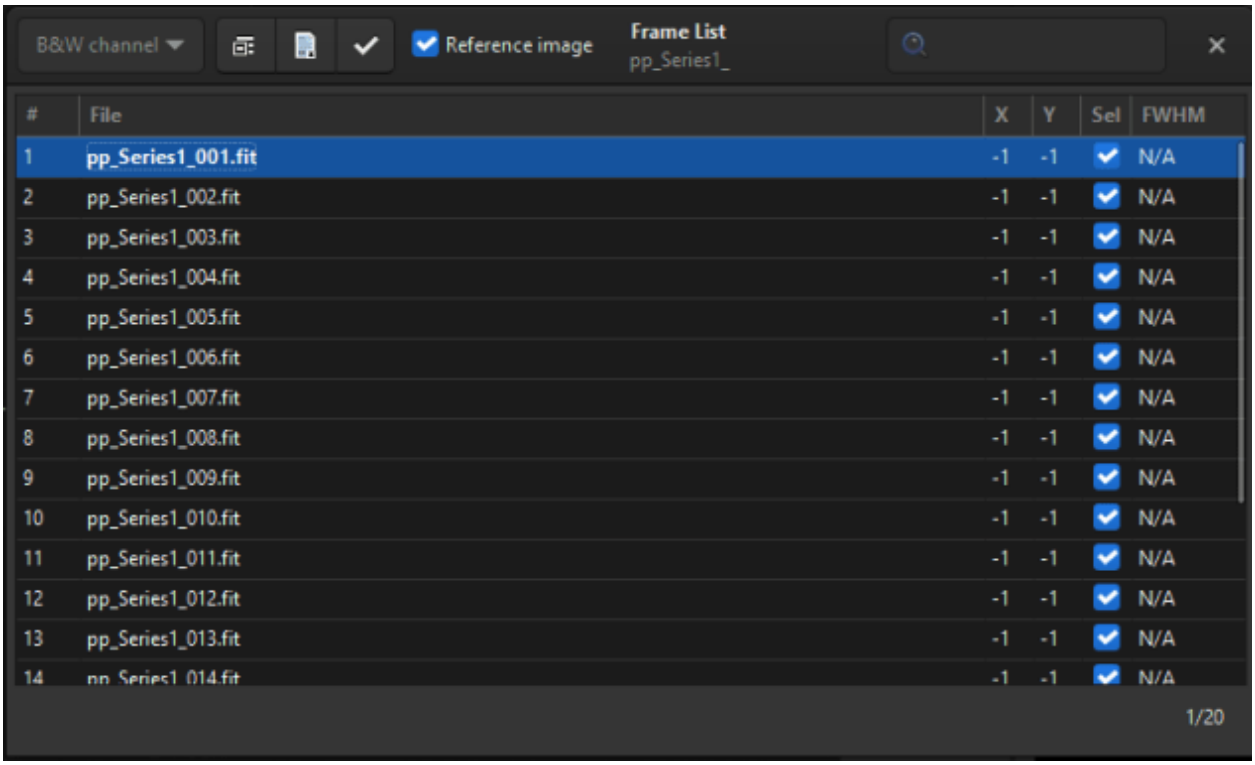


Abb. 22: Das Dialogfeld für die Bildliste. Sie können alle Bilder in der Sequenz durchsuchen.

Beim Stacking wird das Referenzbild auch als Normalisierungsreferenz verwendet, falls die Normalisierung aktiviert ist.

9.3.2 Registrierungsmethoden

Allgemeine Sternausrichtung

Dies ist der bevorzugte Algorithmus zum Ausrichten von Deep-Sky-Bildern mit ausreichender Überlappung.

Die allgemeine Sternausrichtung basiert auf der Methode der Dreiecksähnlichkeit für automatisch erkannte gemeinsamer Sterne in jedem Bild [Valdes1995]. Unsere Implementierung basiert auf dem Programm match von Michael Richmond. Anschließend wird der Algorithmus RANSAC [Fischler1981] auf die Sternlisten angewendet, um weitere Ausreißer auszuschließen und die Projektionsmatrix zu bestimmen. Die Robustheit des Algorithmus hängt von der Fähigkeit ab, die Sterne zu erkennen und gleichzeitig falsche Erkennungen zu vermeiden. Siril verfügt über einen sehr ausgeklügelten Sternerkennungsalgorithmus, der es so weit wie möglich vermeidet, Objekte auszuwählen, die keine Sterne sind, und das in der schnellstmöglichen Zeit. Die Erkennung der hellsten Sterne ist normalerweise am wichtigsten. Wenn jedoch auch schwächere Sterne erkannt werden müssen, kann das Fenster *Dynamische PSF* verwendet werden, um die Erkennungsparameter anzupassen.

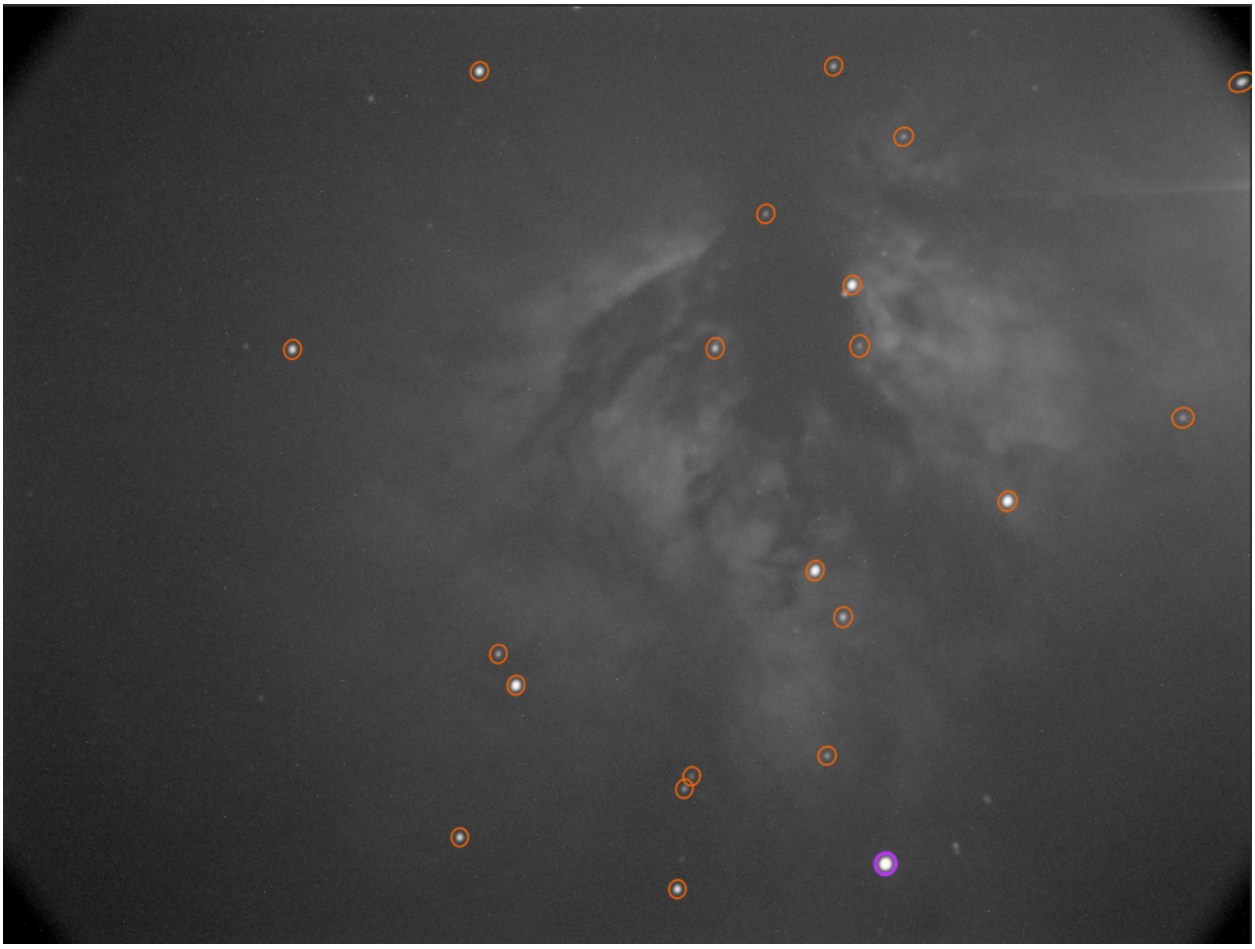


Abb. 23: Automatische Erkennung von Sternen in einem Einzelbild

Es gibt einige Optionen, die mit dieser Ausrichtungsmethode verbunden sind.

Das Dropdown-Menü *Transformation* ermöglicht die Auswahl zwischen verschiedenen Transformationverfahren.

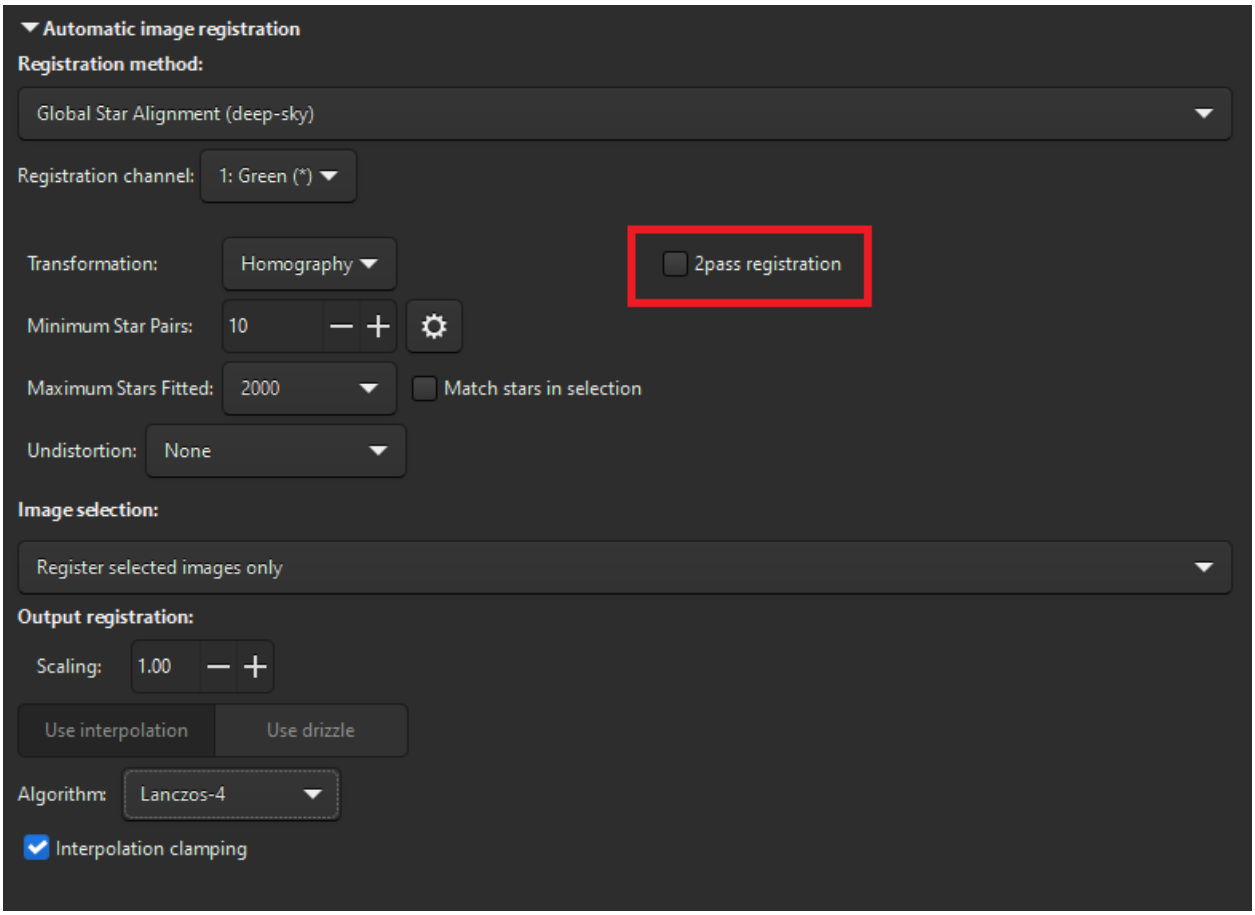


Abb. 24: Globale Registrierungsoptionen

Warnung

Für das anfängliche Stern-Matching wird der Algorithmus der Dreiecksähnlichkeit verwendet, d. h. die Mindestanzahl der Sternpaare muss bei **Verschiebung**, **Ähnlichkeit** und **Affinität** mindestens 3 und bei **Homographie** mindestens 4 betragen.

Andere Registrierungsoptionen sind:

- Die Schaltfläche *Mindest-Stern-Paare* legt die minimale Anzahl von Sternpaaren fest, die ein bestimmtes Bild im Verhältnis zum Referenzbild haben kann. Wenn ein bestimmtes Bild weniger Sternpaare hat, wird es *nicht* registriert. Rechts neben dieser Option befindet sich eine Schaltfläche, die das Werkzeug *Dynamische PSF* öffnet.
- Die Option *Maximale Anzahl von Sternen angepasst* legt die maximale Anzahl der Sterne fest, nach denen in jedem Bild gesucht werden soll (Standardwert 2000). Je größer dieser Wert ist, desto mehr Sterne werden potenziell erkannt, was zu einer längeren Rechenzeit, aber auch zu einer genaueren Registrierung führt.
- Nutze die Option, *Bringe ausgewählte Sterne zur Deckung*, wenn Sie den Global Sternausrichtungs-Algorithmus innerhalb des ausgewählten Bereichs im Referenzbild durchführen möchten. Wenn keine Auswahl aktiv ist, wird diese Option ignoriert.
- Das Dropdown-Menü *Entzerrung* ermöglicht die Auswahl zwischen verschiedenen Korrekturen:
  - Keine
  - Auf Bild
  - Aus FITS/WCS-Datei
  - Von Masterdateien

From image verwendet die astrometrische Lösung aus dem aktuell geladenen Bild.
Aus FITS/WCS-Datei öffnet ein Dateiauswahlfenster, in dem Sie ein FITS-Bild oder eine `*.wcs`-Datei mit Verzerrungskoeffizienten auswählen können.
Von Mastern holt die Master-Verzerrungsdatei, die jedem Bild entspricht. Diese Option kann nützlich sein, wenn Sie Bilder aus zwei oder mehr Sitzungen ausrichten und der Sensor in Bezug auf die Optik verschoben wurde. In diesem Fall sind die Verzerrungskoeffizienten möglicherweise nicht gleich.

Tipp

Damit diese letzte Option *Von Mastern* funktioniert, müssen Sie in den *Einstellungen* einen Master-Verzerrungspfad festgelegt haben. Um eine Masterdatei zu erstellen, gehen Sie *hierhin*.

Mit den Optionen unten können Sie:

- Filtern Sie Bilder heraus, die aus der Sequenz abgewählt wurden.
- Wählen Sie zwischen Interpolation und Drizzle zum Exportieren der Bilder. Diese sind dieselben wie im Abschnitt *Ausgaberegistrierung* und werden hier nicht weiter erläutert.

Siril Kommandozeile

```
register sequencename [-2pass] [-selected] [-prefix=] [-scale=]
register sequencename ... [-layer=] [-transf=] [-minpairs=] [-maxstars=] [-
↪nostarlist] [-disto=]
register sequencename ... [-interp=] [-noclamp]
register sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
```

Sucht und führt optional geometrische Transformationen an Bildern der im Argument angegebenen Sequenz durch, sodass sie auf das Referenzbild gelegt werden können. Dieser Algorithmus verwendet Sterne zur Registrierung und funktioniert nur mit Deep-Sky-Bildern. Optionen zur Sternerkennung können mit **SETFINDSTAR** oder dem Dialog *Dynamische PSF* geändert werden.

Alle Bilder der Sequenz werden registriert, es sei denn, die Option **-selected** wird übergeben, in diesem Fall werden die ausgeschlossenen Bilder nicht verarbeitet

Die Option **-2pass** berechnet nur die Transformationen, generiert aber keine transformierten Bilder. **-2pass** fügt dem Algorithmus einen Vorab-Durchlauf hinzu, um vor der Berechnung der Transformationen basierend auf Bildqualität und Rahmung ein gutes Referenzbild zu finden. Um nach diesem Durchlauf transformierte Bilder zu generieren, verwenden Sie SEQAPPLYREG.

Wenn erstellt, beginnt der Name der Ausgabesequenz mit dem Präfix „r_“, sofern mit der Option **-prefix=** nichts anderes angegeben ist. Die Ausgabebilder können neu skaliert werden, indem ein **-scale=**-Argument mit einem Gleitkommawert zwischen 0,1 und 3 übergeben wird.

Optionen zu Bildtransformationen:

Die Erkennung erfolgt auf der grünen Ebene für Farbbilder, sofern nicht durch die Option **-layer=** mit einem Argument im Bereich von 0 bis 2 für Rot bis Blau etwas anderes angegeben ist.

-transf= gibt die Verwendung von **shift**-, **similarity**-, **affine**- oder **homography**-Transformationen (Standard) an.

Die Option **-minpairs=** gibt die Mindestanzahl von Sternpaaren an, die ein Bild mit dem Referenzbild haben muss, andernfalls wird das Bild ausgelassen und aus der Sequenz ausgeschlossen.

Die Option **-maxstars=** gibt die maximale Anzahl der Sterne an, die in jedem Bild gefunden werden sollen (muss zwischen 100 und 2000 liegen). Mit mehr Sternen kann eine genauere Registrierung berechnet werden, die jedoch mehr Zeit in Anspruch nimmt.

-nostarlist verhindert das Speichern der Sternliste.

-disto= verwendet Verzerrungsterme aus einer vorherigen astrometrischen Lösung (mit einer SIP-Ordnung> 1). Als Parameter wird entweder **image** verwendet, um die im aktuell geladenen Bild enthaltene Lösung zu verwenden, **file** gefolgt vom Pfad zum Bild, das die Lösung enthält, oder **master**, um automatisch den passenden Verzerrungsmaster zu jedem Bild zu laden. Bei Verwendung dieser Option werden die Polynome sowohl zum Korrigieren der Sternpositionen vor der Berechnung der Transformation als auch zum Entzerren der Bilder beim Exportieren der Ausgabebilder verwendet.

Optionen zur Bildtransformation:

Standardmäßig werden Transformationen angewendet, um die Bilder mittels Interpolation zu registrieren.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden.

Optionen zum Drizzeln:

Andernfalls können die Bilder mit dem HST-Drizzle-Algorithmus exportiert werden, indem das Argument **-drizzle** übergeben wird, das die folgenden zusätzlichen Optionen annehmen kann:

-pixfrac= legt den Pixelanteil fest (Standard = 1,0).

Das Argument **-kernel=** legt den Drizzle-Kernel fest und muss von einem der folgenden Argumente gefolgt werden: **point**, **turbo**, **square**, **gaussian**, **lanczos2** oder **lanczos3**. Der Standardwert ist **square**.

Das Argument **-flat=** gibt ein Master-Flat an, um die gedrizzelten Eingabepixel zu gewichten (Standard ist kein Flat).

Hinweis: Wenn Sie **-drizzle** auf Bilder anwenden, die mit einer Farbkamera aufgenommen wurden, dürfen die Eingabebilder nicht debayert werden. In diesem Fall erfolgt die Sternerkennung immer auf den grünen Pixeln

Verweise: *setfindstar*, *psf*, *seqapplyreg*

2-Schritt-Registrierung

Um diese Option zu aktivieren, setzen Sie einfach ein Häkchen in das *entsprechende Kästchen*, nachdem Sie im Methoden-Dropdown-Menü *Globale Registrierung* ausgewählt haben.

Dies führt nur den ersten (von zwei Durchläufen) durch, sodass der Referenzrahmen aus den erkannten Sterninformationen ausgewählt werden kann, anstatt automatisch den ersten Rahmen der Sequenz zu wählen. Die vorgeschlagenen Optionen ähneln dem Algorithmus *Global Registation*, aber diese Methode erstellt keine Sequenzen und alle Ausrichtungsinformationen werden in der Datei `seq` gespeichert.

Bei der Sternerkennung setzt Siril ein Maximum von 2000 zu findenden Sternen (dies kann auch mit der entsprechenden Option geändert werden). Falls mehr als ein Bild die maximale Anzahl an Sternen erreicht hat, werden die Sternlisten aller Bilder erneut durchsucht. Es wird ein neuer minimaler Erkennungsschwellenwert definiert, um die Bilder sowohl nach der Anzahl der erkannten Sterne als auch nach der FWHM sortieren zu können.

Die Sternlisten aller Bilder werden gespeichert, wobei die Erweiterung „.fit(s)“ durch „.lst“ ersetzt wird. Dadurch kann der 2pass-Algorithmus sehr schnell mit anderen Parametern, z. B. einer anderen Transformation, erneut ausgeführt werden. Falls die Sternerkennung geändert wurde, erkennt der Prozess diese Änderungen und führt die Analyse nach Bedarf erneut durch.

Auf diese Registrierung muss im Allgemeinen *Existierende Registrierung anwenden* folgen, um die Transformation anzuwenden und eine neue Sequenz zu erstellen, es sei denn, Sie haben sich für die Berechnung von `Shift` entschieden.

Diese Zeilen führen eine 2-Pass-Registrierung auf eine Sequenz namens *pp_light* durch und wenden sie an. Die Ausgabe ist eine Sequenz *r_pp_light*.

```
# Align lights in 2 passes
register pp_light -2pass
seqapplyreg pp_light
```

Diese Zeilen führen eine 2-Pass-Registrierung auf eine Sequenz namens *colors* durch und wenden sie an, während sie die Ausgabebilder auf den kleinsten gemeinsamen Bereich beschneiden. Die Ausgabe ist eine Sequenz *r_colors*. Dies kann vor dem Zusammensetzen von Monobildern nützlich sein (die Bereiche, die nicht allen Bildern gemeinsam sind, werden abgeschnitten).

```
# Align layers in 2 passes and crop away borders
register colors -2pass
seqapplyreg colors -framing=min
```

1-2-3 Stern-Registrierung

Wenn die Bilder nur wenige Sterne enthalten, z. B. bei DSO Lucky Imaging-Bildern mit einer Belichtungszeit von weniger als einer Sekunde, ist es möglich, dass der globale Registrierungsalgorithmus fehlschlägt, selbst wenn Sie die Erkennungsparameter im Fenster *Dynamische PSF* ändern. In diesem Fall kann es interessant sein, die Sterne, die Sie ausrichten möchten, manuell zu erkennen. Dies ist der Zweck des Algorithmus zur Registrierung von 1, 2 oder 3 Sternen.

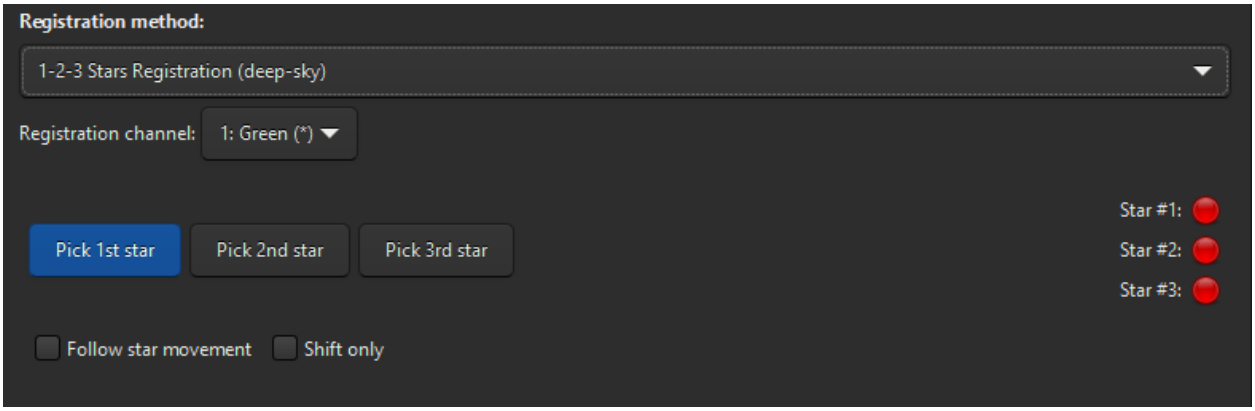


Abb. 25: 1-2-3 Stern-Registrierungsoptionen

Das Prinzip dieser Methode besteht darin, einen Auswahlbereich um einen Stern zu ziehen und auf die Schaltfläche *1. Stern auswählen* zu klicken, dann entsprechend mit dem zweiten und dritten Stern.

- Wenn nur ein Stern ausgewählt ist, wird nur die Verschiebung zwischen den Bildern berechnet. Daher wird die Schaltfläche *Nur verschieben* automatisch ausgewählt. Die Verschiebungswerte werden dann in der Datei `seq` gespeichert.
- Wenn zwei oder drei Sterne ausgewählt werden, kann die Drehung berechnet und angewendet werden, um eine neue Sequenz zu erstellen. Wenn jedoch die Option *Nur Verschiebung* gewählt wird, was nicht zwingend erforderlich ist, wird nur die Verschiebung berechnet.

Die Option *Folge der Sternbewegung* verwendet die Position des/der im vorherigen Bild gefundenen Sterns/Sterne als neues Zentrum für die aktuelle Bildregistrierung. Dadurch wird der Auswahlbereich kleiner, die Registrierung schneller und berücksichtigt Drift oder Bilder mit einer großen Anzahl von Sternen.

> **Warnung**
>
> Wenn Sie diese Option aktivieren, wird die Registrierung nicht parallelisiert, sondern nur auf einem CPU-Kern ausgeführt.

Bildmusterausrichtung

Hierbei handelt es sich um ein einfaches Verfahren zur Registrierung durch Verwendung der Kreuzkorrelation <https://de.wikipedia.org/wiki/Kreuzkorrelation>`_ im räumlichen Bereich.

Diese Methode ist schnell und wird für die Registrierung von *Planetenvideos* verwendet, bei denen in großen Bereichen des Bildes zusammenhängende Informationen zu sehen sind. Sie kann auch für die Registrierung einiger Deep-Sky-Bilder verwendet werden. Beachten Sie jedoch, dass es sich dabei um eine Ein-Punkt-Ausrichtungsmethode handelt, die für die Ausrichtung von hochauflösenden Planetenbildern schlecht geeignet ist. Aber es registriert die Bilder effektiv, um die Sequenz zu stabilisieren. Zeichnen Sie einfach eine Auswahl um das Objekt (z. B. den Planeten) und stellen Sie sicher, dass seine Bewegung während der Sequenz innerhalb der Auswahl bleibt Mit dieser Methode kann nur die Verschiebung berechnet werden.

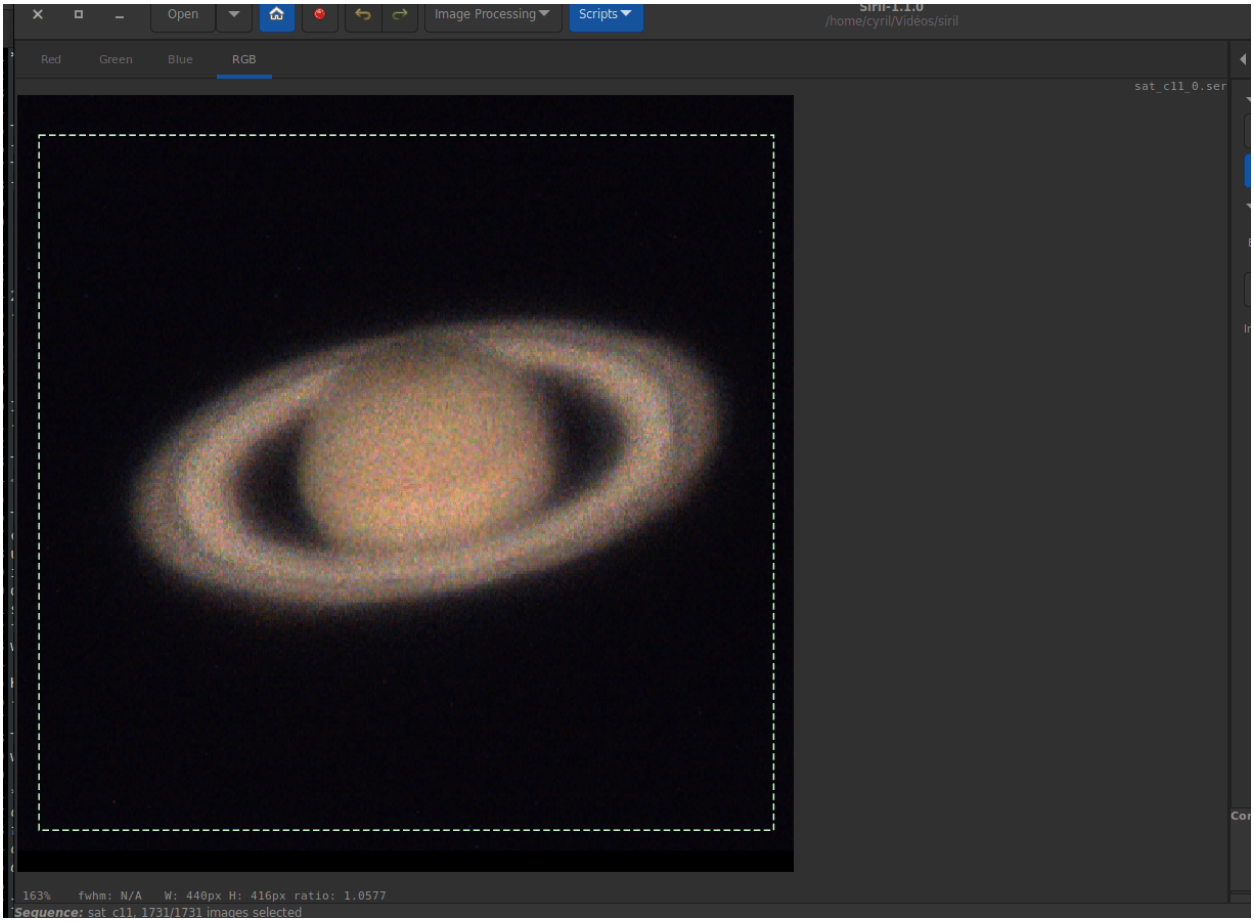


Abb. 26: Musterausrichtungsoptionen

KOMBAT

Diese Methode stammt aus der Bibliothek OpenCV, einer in Siril häufig genutzten Bibliothek. Sie erklären:

Es schiebt einfach das Referenzbild über das Eingabebild (wie bei der 2D-Konvolution) und vergleicht die Vorlage und den Bereich des Eingangsbildes unter dem Referenzbild. Mehrere Vergleichsmethoden sind in OpenCV implementiert. (Sie können die Dokumentation für weitere Details einsehen). Das Ergebnis ist ein Graustufenbild, bei dem jedes Pixel angibt, wie sehr die Umgebung dieses Pixels mit der Vorlage übereinstimmt.

In der Praxis zeichnen Sie einfach eine Auswahl um das Objekt (z. B. den Planeten) und stellen sicher, dass seine Bewegung während der Sequenz innerhalb der Auswahl bleibt. Mit dieser Methode kann nur die Verschiebung berechnet werden.

Kometen-/Asteroiden-Registrierung

Das Kometenregistrierungstool funktioniert auf sehr einfache Weise in zwei Schritten.

1. Wählen Sie mit der Bilderliste das erste Bild der Sequenz, umgeben Sie den Kometenkern und klicken Sie dann auf die Schaltfläche *Wähle ein Objekt in #1*.
2. Wählen Sie dann das letzte Bild der Sequenz aus, wählen Sie den Kern des Kometen aus und klicken Sie auf die Schaltfläche *Wähle ein Objekt in #2*.

Die Kometengeschwindigkeit $Deltax$ und $Deltay$ wird in Pixel pro Stunde berechnet, wenn alles in funktioniert hat.

Warnung

Die Ausrichtung des Kometen muss an Bildern vorgenommen werden, deren Sterne zuvor ausgerichtet wurden. Entweder über eine neue Sequenz, mit der globalen Sternausrichtung, oder indem man die Registrierungsinformationen über eine 2-Pass- oder astrometrische Registrierung in der Datei `seq` gespeichert hat.

Bemerkung

Um voll funktionsfähig zu sein, müssen die Bilder einen Zeitstempel haben.

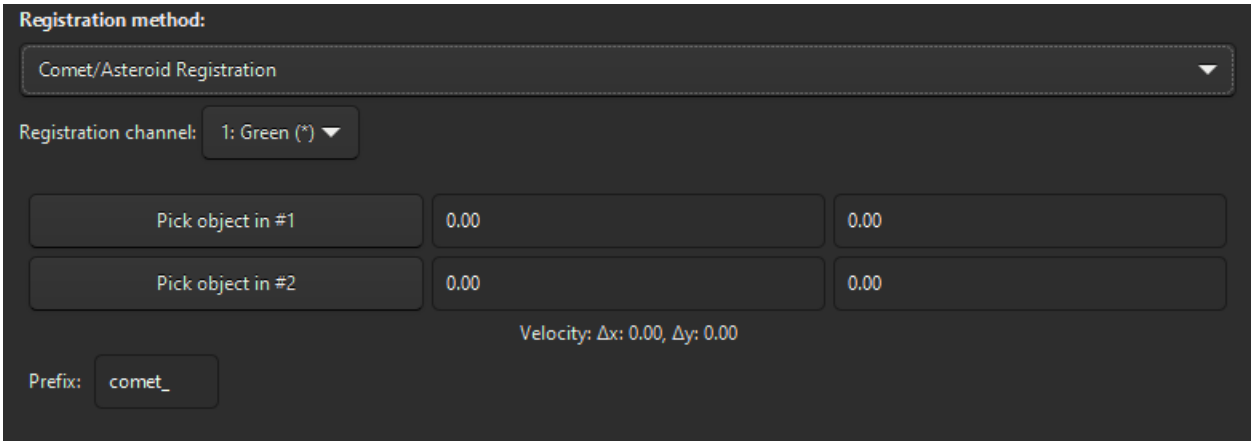


Abb. 27: Kometenregistrierungsoptionen

Tipp

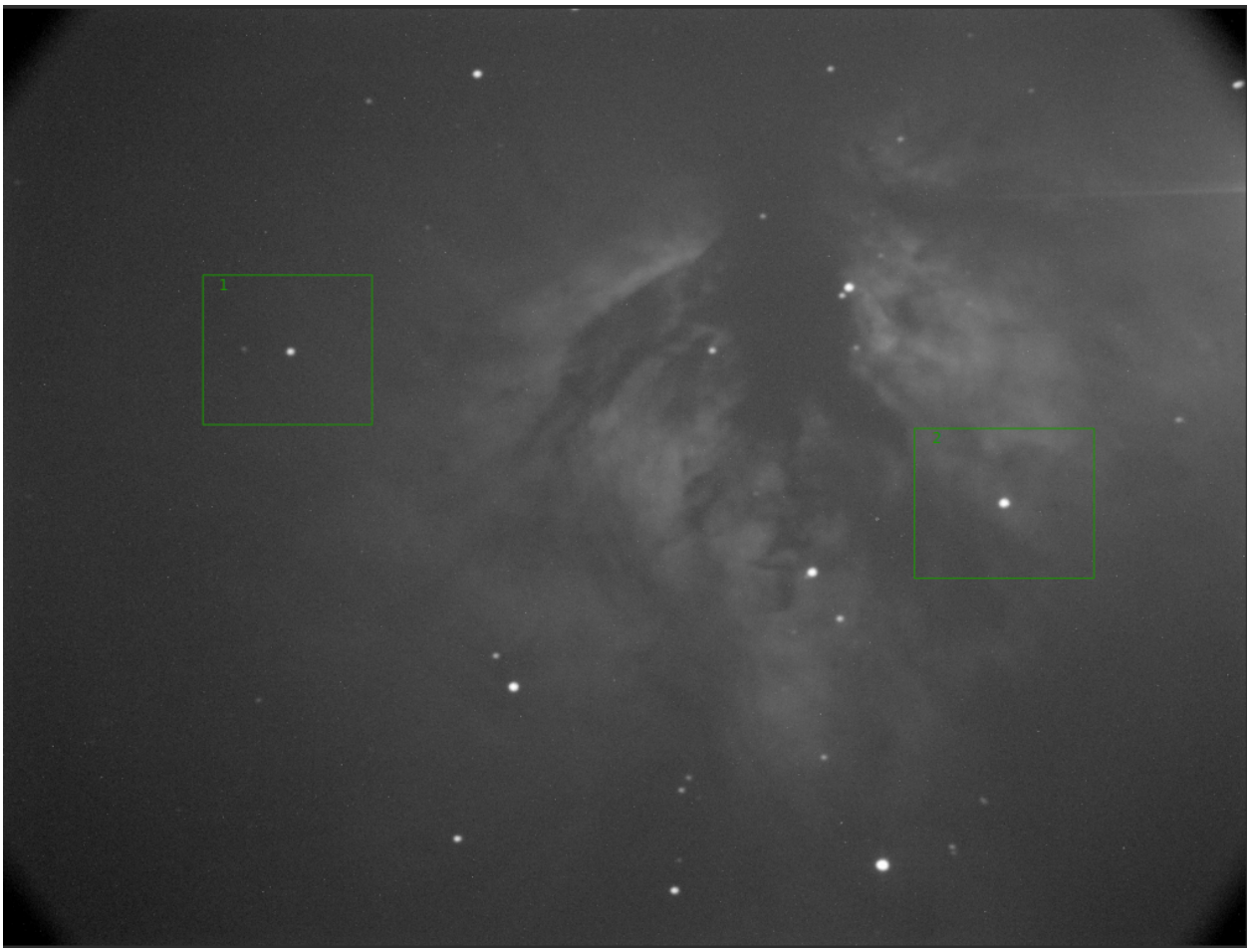
Wenn die PSF-Erkennung ein Objekt nicht erkennt, wird die Mitte des gezeichneten Kästchens zurückgegeben. Dies kann praktisch sein, wenn Sie sich auf ein Objekt ausrichten möchten, das auf Ihren Einzelbildern nicht sichtbar ist. Verwenden Sie die Beschriftung *Solarsystemobjekt*, um die Position eines Asteroiden zu markieren, und zeichnen Sie ein Kästchen um die Markierung, um seine Position auszuwählen.

Diese Methode gibt eine neue Sequenzdatei aus, wobei dem Sequenznamen das definierte *prefix* vorangestellt wird (standardmäßig `comet_`). Dadurch werden jedoch keine neuen Bilder erstellt, sondern symbolische Links zu den Originalbildern. Windows-Benutzer müssen sicherstellen, dass sie den Entwicklermodus aktiviert haben, da sonst echte Kopien erstellt werden. Siehe auch den Hinweis unten *diese Tabelle*.

Manuelle Registrierung

Diese letzte Methode der Registrierung ist sehr speziell, was ihre separate Position erklärt, und erlaubt es, Bilder manuell auszurichten. Natürlich ist nur die Verschiebung zwischen Bildern erlaubt.

Als erstes müssen Sie zwei Vorschaubilder im Bild definieren. Durch Anklicken der Schaltfläche *Definieren Sie die erste Vorschau* wird die erste Vorschau initialisiert. Sie müssen dann auf einen Bereich des Bildes klicken, idealerweise auf einen Stern in der Nähe eines Bildrandes, um den Vorschaubereich festzulegen. Ein Klick auf die zweite Schaltfläche *Definieren Sie die zweite Vorschau* ermöglicht es, das Gleiche für einen zweiten Punkt zu tun.



Es ist sehr wichtig, dass Sie bereits ein Referenzbild mit der *Bilderliste* festgelegt haben. Standardmäßig ist dies das erste

Bild. Der Benutzer kann das gewünschte Bild frei wählen. Es wird als Referenzebene verwendet, die durch Transparenz sichtbar ist, um die Bilder manuell mit den Verschiebungs-Schaltflächen auszurichten. Anschließend können Sie die Bilder einzeln durcharbeiten, um die gleiche Methode auf die gesamte Sequenz anzuwenden.

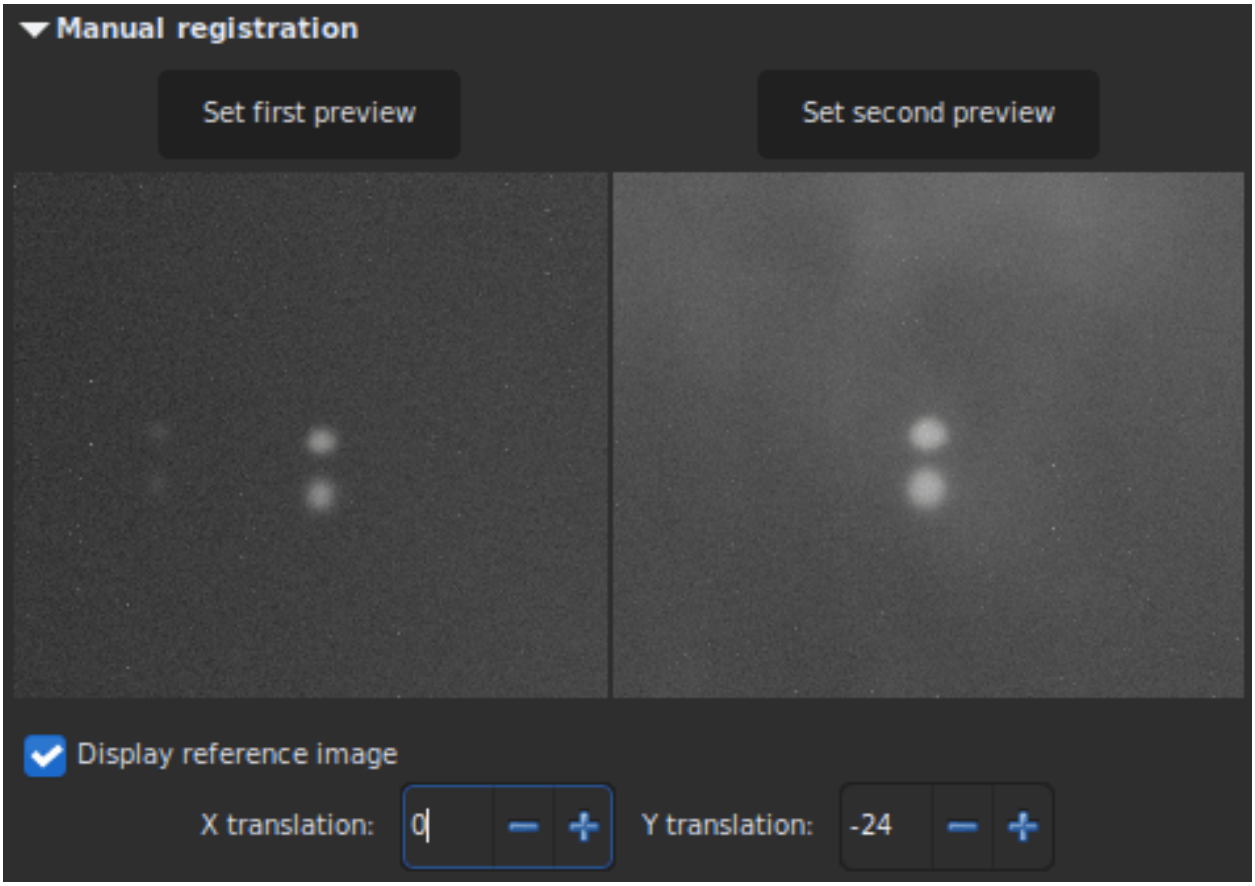


Abb. 28: Die Y-Verschiebung ist zu groß, gleiche Sterne auf verschiedenen Bildern überschneiden sich nicht.

Astrometrische Registrierung

Dieser in Version 1.3 eingeführte Modus ist der bevorzugte zum Zusammenstellen von Mosaiken oder Bildern mit geringer Überlappung. Er kann auch nützlich sein, um Stapel zu registrieren, die aus unterschiedlichen Konfigurationen stammen (unterschiedliche Optiken, unterschiedliche Kameras, unterschiedliche Sichtfelder usw.).

Es gibt keinen Eintrag im Dropdown-Menü der Registrierungsmethode, da die Informationen zum Exportieren der registrierten Bilder bereits bei der *astrometrischen Lösung der Sequenz* berechnet wurden. Sie müssen lediglich die *vorhandene Registrierung anwenden* auswählen.

Die Entzerrung wird wie beim astrometrischen Lösen der Sequenz definiert angewendet, d. h. wenn die Bilder mit einer SIP-Ordnung größer als 1 gelöst wurden, wird die Entzerrung automatisch einbezogen. Sofern Sie kein optisch perfekt flaches Feld haben, ist es normalerweise eine gute Idee, mit SIP zu lösen, wie unten gezeigt, mit und ohne Verzerrungskorrektur.

Vorhandene Registrierung anwenden

Dabei handelt es sich nicht um einen Algorithmus, sondern vielmehr um ein Hilfsmittel zur Anwendung zuvor berechneter Registrierungsdaten, die in der Sequenzdatei gespeichert sind. Die Interpolationsmethode kann im Abschnitt Registrierungsausgabe ausgewählt werden. Sie können auch die Bildfilterung verwenden, um zu vermeiden, dass un-

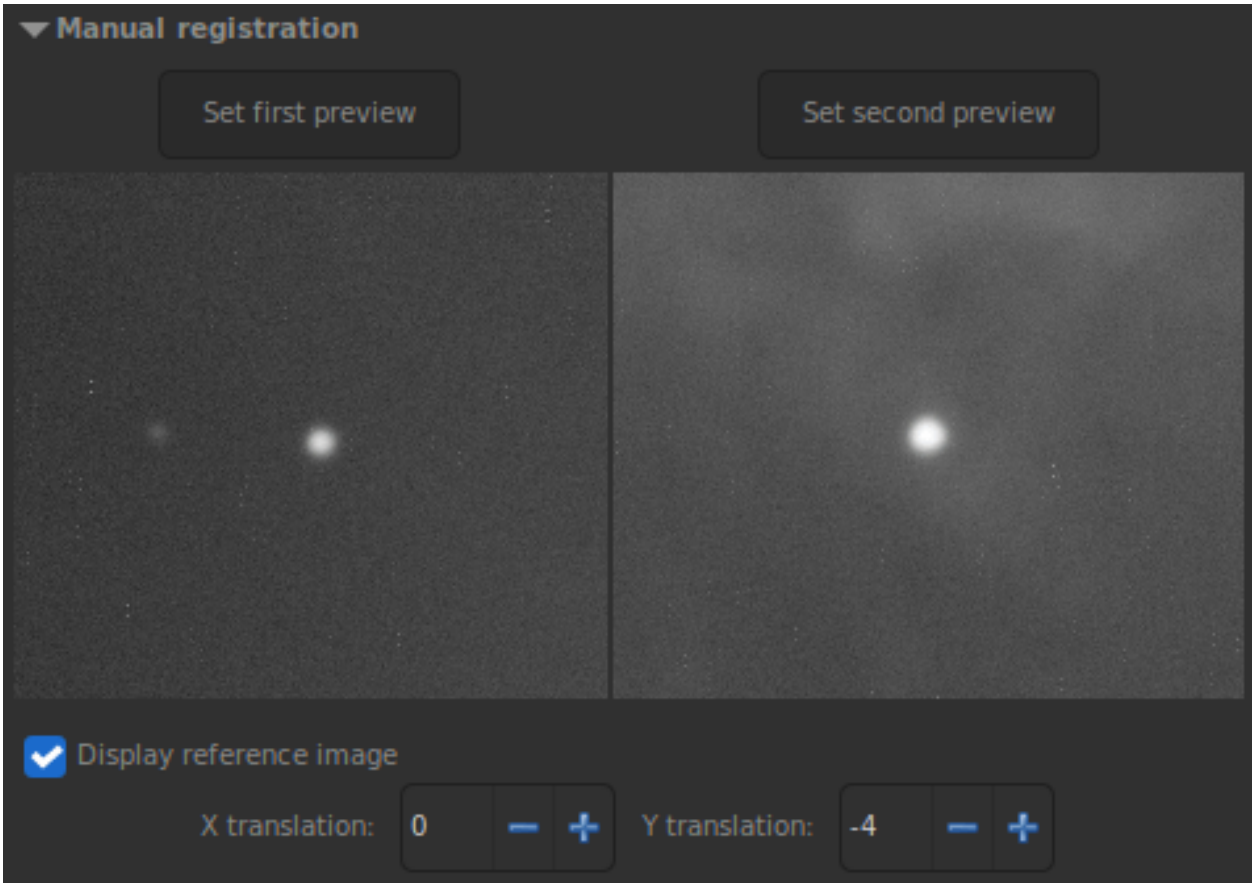


Abb. 29: X- und Y-Verschiebung sehen gut aus. Das aktuelle Bild ist am Referenzbild ausgerichtet.

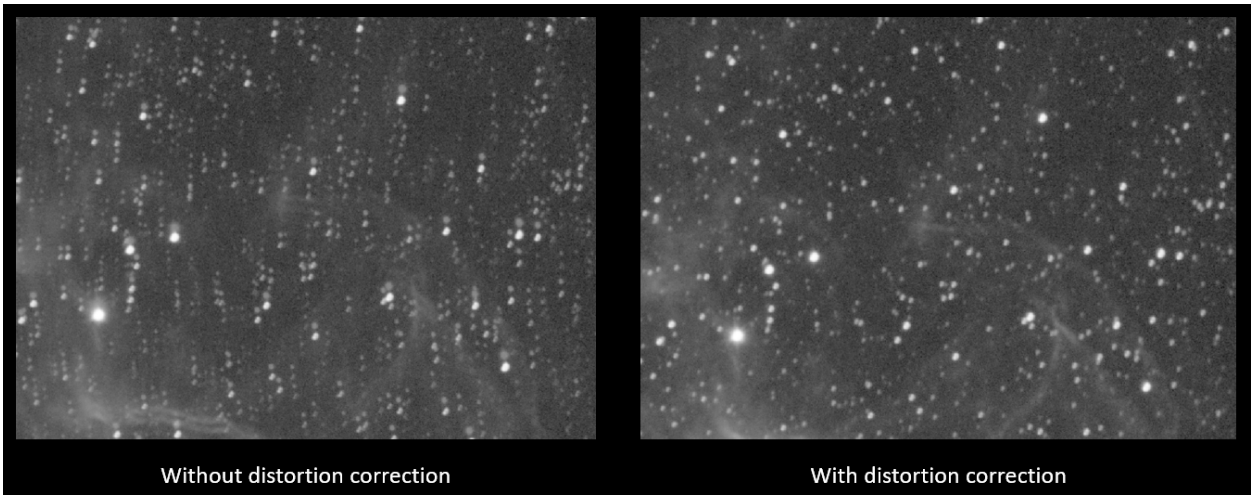
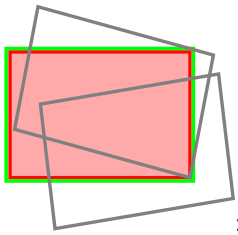
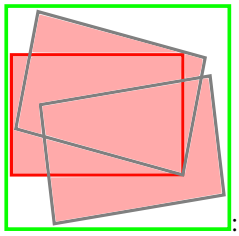
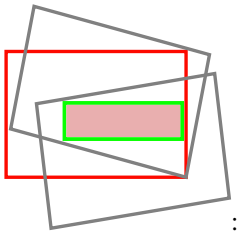
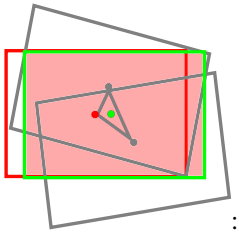


Abb. 30: Auswirkung der Verzerrungskorrektur auf zwei sich überlappende Bilder nach der Registrierung

nötige Bilder gespeichert werden, wie beim Stacking *Bilder ausschließen*. Es gibt auch eine *Drizzle*-Option, um die Registrierung mit Drizzle anstelle von Interpolation durchzuführen. Siehe *Drizzle Abschnitt* für Details.

Es stehen vier Rahmungsmethoden zur Verfügung:

-  : **aktuell** verwendet das aktuelle Referenzbild. Dies ist das Standardverhalten.
-  : **Maximum** (Bounding Box) fügt bei Bedarf einen schwarzen Rand um jedes Bild hinzu, damit kein Teil des Bildes beim Registrieren abgeschnitten wird.
-  : **Minimum** (gemeinsamer Bereich) beschneidet jedes Bild auf den Bereich, den es mit allen Bildern der Sequenz gemeinsam hat.
-  : **Schwerpunkt** bestimmt die beste Position für den Bildausschnitt als Schwerpunkt (cog, Center Of Gravity) aller Bilder.

> **Tipp**
>
> Der in Siril 1.3 eingeführte „max“-Modus exportiert keine Bilder mit schwarzen Rändern, die das gesamte resultierende Bild umfassen. Die Bilder werden mit der erforderlichen Projektion exportiert und die relativen Verschiebungen, die zum Erstellen des endgültigen Bildes erforderlich sind, bleiben in der resultierenden Sequenzdatei erhalten.

Button *Estimate*, der die Berechnung des Rahmens startet, ohne die Bilder tatsächlich zu exportieren. Diese Information kann interessant sein, um im Voraus die Größe der exportierten Bilder zu kennen. Dies berücksichtigt die gewählte **Rahmungsmethode** und den **Skalierungsfaktor**, der in der *Output-Registrierung* gewählt wurde

Wenn Sie *Estimate* (Abschätzen) drücken, zeigt die Konsole eine Ausgabe wie diese:

```
Output image: 7893 x 5254 pixels (assuming a scaling factor of 1.30)
```

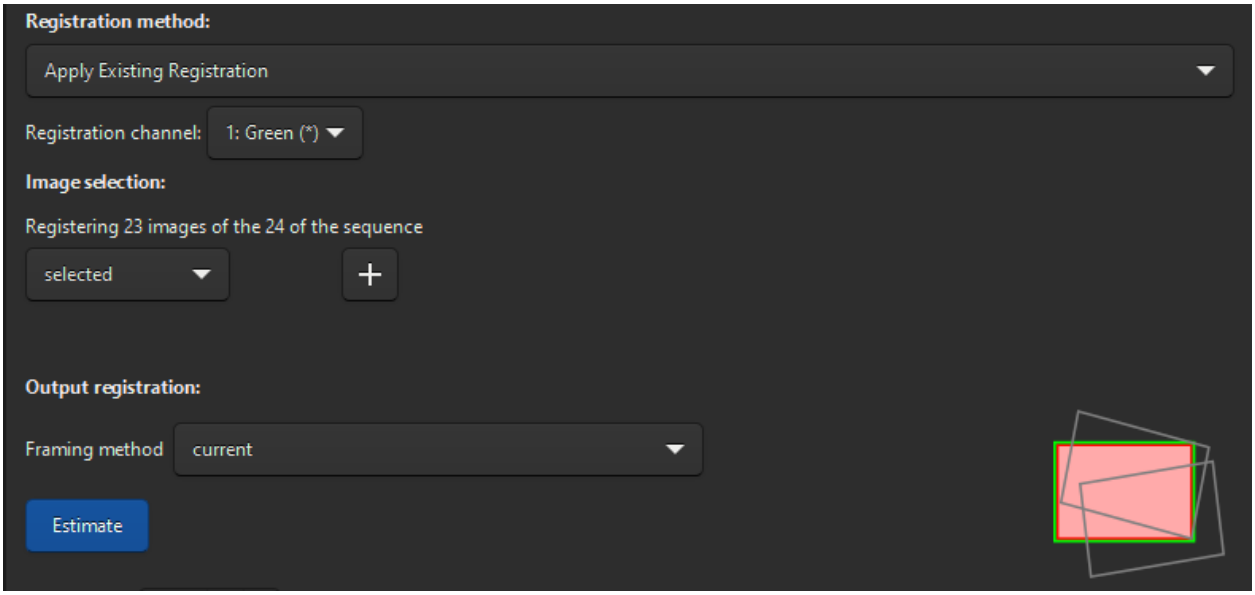


Abb. 31: Vorhandene Registrierungsoptionen anwenden

Siril Kommandozeile

```
seqapplyreg sequencename [-prefix=] [-scale=] [-layer=] [-framing=]
seqapplyreg sequencename ... [-interp=] [-noclamp]
seqapplyreg sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
seqapplyreg sequencename ... [-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-
↪filter-round=value[%|k]] [-filter-bkg=value[%|k]] [-filter-nbstars=value[%|k]] [-
↪filter-quality=value[%|k]] [-filter-incl[uded]]
```

Wendet geometrische Transformationen auf Bilder der im Argument angegebenen Sequenz an, so dass sie dem Referenzbild überlagert werden können, wobei zuvor berechnete Registrierungsdaten verwendet werden (siehe REGISTER).

Der Name der Ausgabesequenz beginnt mit dem Präfix **"r_"**, sofern mit der Option **-prefix=** nichts anderes angegeben wurde.

Bei RGB-Bildern wird die Registrierung auf der ersten Ebene durchgeführt, für die Daten vorhanden sind, es sei denn, die Option **-layer=** (0, 1 oder 2 für R, G bzw. B) wurde angegeben.

Die ausgegebenen Bilder können durch Übergabe eines Wertes für das Argument **-scale=** (Skalierung, zwischen 0,1 und 3) vergrößert oder verkleinert werden.

Die automatische Beschneidung der Ausgabesequenz kann mit dem Schlüsselwort **-framing=**, gefolgt von einer der Methoden aus der Liste { current | min | max | cog } festgelegt werden:

-framing=max (Begrenzungsrahmen) projiziert jedes Bild und berechnet seine Verschiebung in Bezug auf das Referenzbild. Die resultierende Sequenz kann dann mit der Option **-maximize** des STACK-Befehls gestackt werden, wodurch das vollständige Bild erstellt wird, das alle Bilder der Sequenz umfasst.

-framing=min (gemeinsamer Bereich) beschneidet jedes Bild auf den Bereich, den es mit allen Bildern der Sequenz gemeinsam hat.

-framing=cog bestimmt die beste Position für den Bildausschnitt als Schwerpunkt (cog) aller Bilder.

Optionen zur Bildtransformation:

Standardmäßig werden Transformationen angewendet, um die Bilder mittels Interpolation zu registrieren.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden.

Optionen zum Drizzeln:

Andernfalls können die Bilder mit dem HST-Drizzle-Algorithmus exportiert werden, indem das Argument **-drizzle** übergeben wird, das die folgenden zusätzlichen Optionen annehmen kann:

-pixfrac= legt den Pixelanteil fest (Standard = 1,0).

Das Argument **-kernel=** legt den Drizzle-Kernel fest und muss von einem der folgenden Argumente gefolgt werden: **point**, **turbo**, **square**, **gaussian**, **lanczos2** oder **lanczos3**. Der Standardwert ist **square**.

Das Argument **-flat=** gibt ein Master-Flat an, um die gedrizzelten Eingabepixel zu gewichten (Standard ist kein Flat).

Bilder herausfiltern:

Die zu registrierenden Bilder können auf der Grundlage einiger Filter ausgewählt werden (z. B. derjenigen mit der besten FWHM), mit einigen der **-filter-*** Optionen.

Links: *register*, *stack*

Einige davon werden gefiltert, ohne dass eine bestimmte Reihenfolge oder Anzahl festgelegt wird:

```
[-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-filter-round=value[%|k]] [-
↪filter-bkg=value[%|k]]
[-filter-nbstars=value[%|k]] [-filter-quality=value[%|k]] [-filter-incl[uded]]
```

Die besten Bilder der Sequenz können mit Hilfe der Filterargumente gestackt werden. Jedes dieser Argumente kann schlechte Bilder auf der Grundlage einer Eigenschaft, die ihrem Namen entpricht, aus den Registrierungsdaten mit einem der drei Typen von Argumentwerten entfernen:

- a numeric value for the worse image to keep depending on the type of data used (between 0 and 1 for roundness and quality, absolute values otherwise),

- a percentage of best images to keep if the number is followed by a % sign,

- or a k value for the k.sigma of the worse image to keep if the number is followed by a k sign.

Es ist auch möglich, manuell ausgewählte Bilder zu verwenden, entweder zuvor über die grafische Benutzeroberfläche oder mit den Befehlen select oder unselect, unter Verwendung des Arguments **-filter-included**.

9.3.3 Ausgabe Registrierung

Dieser Bereich enthält alle Ausgabeelemente für die Sequenz. Sie können zwischen Interpolation und Drizzle wählen, um die Bilder zu exportieren.

> **Warnung**
>
> Abhängig von der Art der Eingabesequenz sind möglicherweise nicht beide verfügbar:
>
> - für Monosequenzen können sowohl Interpolation als auch Drizzle ausgewählt werden
> - für CFA-Sequenzen (undebayerte Farbbilder) ist nur Drizzle verfügbar
> - für RGB-Sequenzen (debayerte Farbbilder) ist nur Interpolation verfügbar

Beide Methoden teilen diese Optionen:

- :guilabel:`Scaling`kann zwischen 0,1 und 2 eingestellt werden, um die Größe der ausgegebenen Bilder zu skalieren.
- *Präfix*, das Präfix, das dem Namen der exportierten Sequenz vorangestellt wird (standardmäßig `r_`).

Interpolation

Die Pixel der resultierenden Bilder werden durch einen Algorithmus interpoliert, der vom Benutzer ausgewählt werden kann. Es gibt 5 mögliche Interpolationsalgorithmen sowie eine Option **Keine**:

- Nächster Nachbar
- Bilinear
- Bikubisch
- Pixel zu Fläche Verhältnis
- Lanczos-4
- Keine

Die effizientesten Interpolationsmethoden sind im Allgemeinen bikubisch und Lanczos-4 (standardmäßig verwendet). Sie erfordern jedoch in der Regel die Aktivierung der Option *Interpolation* mit der eine Begrenzung der Interpolation aktiviert wird, um Ring-Artefakte um die Sterne zu vermeiden. Wir aktivieren diese Option weiterhin standardmäßig, da sie eine deutliche Verbesserung der Auflösung bewirkt, mit einer um 8 % geringeren FWHM im Vergleich zur Standardmethode von Siril 1.0 für den Pixelbereich. Es ist möglich, dass die Begrenzung für Ihre Bilder nicht sinnvoll ist. Wir empfehlen Ihnen, die Ergebnisse mit Ihren Bildern zu vergleichen.

Der Sonderfall **Keine** ist für den Fall der globalen Registrierung und Vorhandene Registrierung anwenden reserviert. Wenn Sie eine Sequenz exportieren oder speichern möchten, die nur eine Verschiebung enthält, ohne eine Interpolation zu verwenden (um die Pixelwerte nicht zu verändern), sollten Sie **Keine** wählen.

Drizzle

Die Schaltfläche *Drizzle* aktiviert den Drizzle-Algorithmus für die Verarbeitung dieser Sequenz. Siehe *Drizzle Abschnitt* für Details.

> **Warnung**
>
> Der Nachteil dieser Technik ist, dass die Menge an Speicher und Festplattenplatz, die für die Erstellung und Verarbeitung von Drizzle-Bildern benötigt wird, mit dem Quadrat des Drizzle-Skalierungsfaktors größer wird.

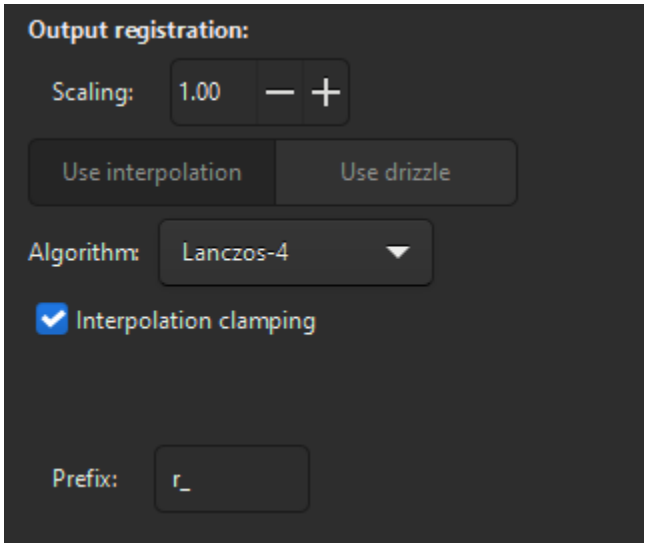


Abb. 32: Ausgabeoptionen für die Interpolation

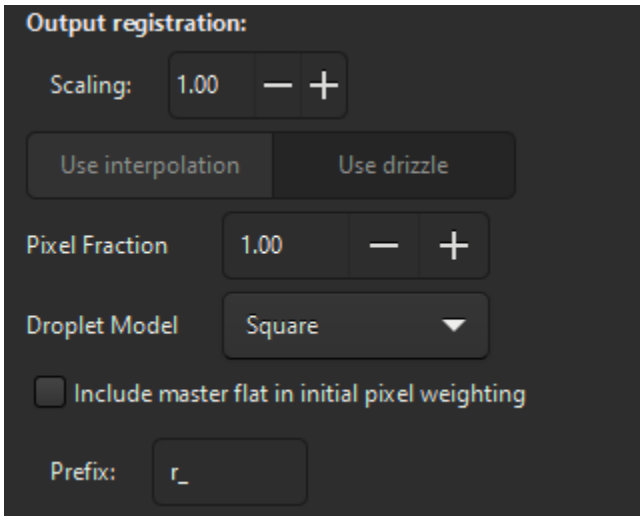


Abb. 33: Ausgabeoptionen für Drizzle

Astrometrische Lösung der registrierten Bilder

Wenn registrierte Bilder exportiert werden, übernehmen sie die astrometrische Lösung des Referenzbildes, sofern vorhanden. Andernfalls wird ihre vorherige Lösung gelöscht. Die neue Lösung für jedes Bild berücksichtigt natürlich die Transformationen, die es während des Registrierungsprozesses erfahren hat.

Falls im Referenzbild eine Verzerrungslösung gefunden wird, aber beim Berechnen der Transformationen keine Entzerrung angewendet wurde – beispielsweise wenn Sie in globalen oder 2-Pass-Methoden keine „Verzerrung“ aus der Dropdown-Liste ausgewählt haben – wird in der Konsole eine Warnung angezeigt. Die Verzerrungsinformationen bleiben in den registrierten Bildern erhalten. Falls sie erheblich sind, können Sie ihre Auswirkung beim Stacking sehen. In diesem Fall müssen Sie sich erneut mit einer Verzerrungsspezifikation registrieren.

9.3.4 Quellenverzeichnis

9.4 Drizzle

Die lineare Rekonstruktion mit variablen Pixeln, besser bekannt als Drizzle, wurde von Andy Fruchter und Richard Hook [FruchterHook1997] für die Arbeit am Hubble-Weltraumteleskop (HST) entwickelt, zunächst zum Zweck der Kombination geditherter Bilder des Hubble Deep Field North (HDF-N). Man kann sich diesen Algorithmus als eine kontinuierliche Reihe von linearen Funktionen vorstellen, die gleichmäßig zwischen der optimalen linearen Kombinationstechnik (Interlacing) und Shift-and-Add variieren. Dies ermöglicht oft eine Verbesserung der Auflösung und eine Verringerung des korrelierten Rauschens im Vergleich zu Bildern, die nur mit Shift-and-Add erzeugt werden.

Es gibt eine ausgezeichnete Seite, die die Technik zusammenfasst und eine gute grafische Darstellung der Art und Weise liefert, wie die Pixeldaten vom groben Eingabepixelraster auf ein feineres Ausgabepixelraster "herabrieseln" - hier <https://www.stsci.edu/~fruchter/dither/drizzle.html>`_.

Das Ausmaß, in dem der Algorithmus vom Zeilensprungverfahren abweicht und sich in Richtung Shift-and-Add bewegt, hängt davon ab, wie gut die PSF durch die Verschiebungen in den Eingangsbildern unterabgetastet wird. In der Praxis wird das Verhalten des Drizzle-Algorithmus durch die Verwendung eines Parameters gesteuert, der als Pixelanteil bezeichnet wird und den Betrag darstellt, um den die Eingangspixel geschrumpft werden, bevor sie auf die Ausgabebildebene abgebildet werden. Bei einem Pixelanteil von 0 entspricht Drizzle dem reinen Zeilensprungverfahren; bei einem Pixelanteil von 1 entspricht es dem Verschieben und Hinzufügen.

Um den Unterschied zwischen der Drizzle- und der Interpolationsmethode bei der Anwendung der Registrierung zu verstehen, betrachten wir zunächst, wie die Standardinterpolationsmethode funktioniert. Die Registrierungsdaten haben die Form einer 3x3-Homographie-Matrix, die eine lineare Transformation mit 8 Freiheitsgraden von einem Koordinatensatz zum anderen (d. h. von jedem Bild zum Referenzbild) kodiert. Diese wird verwendet, um die Werte jedes Pixels in jedem Eingabebild auf die richtige Stelle im Ausgabebild abzubilden und die Ausgabe am Referenzbild auszurichten. Für die eigentliche Ausrichtung wird eine Interpolationsmethode verwendet, die in den Registrierungsoptionen ausgewählt werden kann. Die Interpolation führt zu einer Verschmierung der Punktspreizfunktion, insbesondere beim Hochskalieren von Bildern. Sie kann auch zu Artefakten führen, obwohl Siril einen Klammermechanismus implementiert, um dies zu minimieren.

Bei Drizzle hingegen wird jedes Pixel des Eingangsbildes in ein Tröpfchen verwandelt und durch ein Gittermuster auf das Ausgangsreferenzbild geträufelt. Jedes Tröpfchen hat eine bestimmte Größe, und durch die Wahl eines vergrößerten Ausgabepixelrasters, aber einer kleineren Tröpfchengröße, können Sie eine bessere Auflösung erzielen, **wenn** Ihr Image Train unterabgetastet ist. (Wenn die Abtastung für das Auflösungsvermögen Ihres Teleskops korrekt ist, kann Drizzle keine Details jenseits der Beugungsgrenze erzeugen). Der Preis dafür ist ein erhöhtes Bildrauschen: Da jedes Tröpfchen einen kleineren Bereich im Ausgabebild "malt", wird die durchschnittliche Anzahl der Tröpfchen pro Ausgabepixel im endgültigen Stapel reduziert.

Beachten Sie, dass Drizzle nicht den gesamten Registrierungsprozess ersetzt: Sie verwenden weiterhin die globale Sternausrichtung, die 1-2-3-Sternausrichtung, die Kometenregistrierung oder eine andere Registrierungsmethode, die Sie vor dem Drizzeln anwenden möchten: Es ist lediglich eine Alternative zur Interpolationsmethode, die bei der Anwendung der Registrierung verwendet wird.

Warnung

Beim Drizzle-Prozess werden mehrere Frames genommen und zu einem Ausgabe-Frame zusammengefügt. Die Eingabe ist der Frame-Satz und ihre WCS-Lösungen, und die Ausgabe ist das einzelne zusammengefügte Bild. In Siril wird der Prozess in Drizzle-Registrierung und anschließendes Stacken aufgeteilt, das Zwischenartefakt (der Satz von Bildern, der die einzelnen Bilder darstellt, die auf das gemeinsame Ausgaberaster zusammengefügt, aber noch nicht gestackt wurden) ist jedoch nichts, das an sich von Bedeutung ist. Einzelne Frames in der zusammengefügten Sequenz können **und werden** seltsam aussehen: Der Referenz-Frame sieht oft anders aus als die anderen, insbesondere beim Zusammenfügen von CFA-Daten, aufgrund seiner besonderen Eigenschaft, keine Verschiebung in Bezug auf den Ausgabe-Frame zu haben, und andere Frames können Moiré-Muster aufweisen. **Keine Panik!** Dies ist lediglich ein Zwischenschritt im gesamten Drizzle-Prozess, und all diese scheinbaren Merkwürdigkeiten werden beim Stacking behoben.

9.4.1 Drizzle benutzen

Es gibt 3 Hauptgründe, warum Sie Drizzle einer Interpolationsmethode für die Registrierung vorziehen.

- **Auflösungsverbesserung** Wenn Ihr Bild stark unterabgetastet (undersampled) ist, können Sie mit Drizzle möglicherweise eine Auflösungsverbesserung erzielen, die Sie mit der Registrierungsoption *Interpolation x2 hochskalieren* nicht erreichen können.
- **CFA-Bilder** Wenn Ihre Bilder ein CFA-Muster aufweisen (d. h. wenn Sie eine One-Shot-Color-Kamera oder eine DSLR verwenden), bietet Drizzle eine erhebliche Verbesserung gegenüber dem Debayering. Dies wird manchmal getrennt von Drizzle als *Bayer Drizzle* bezeichnet, aber eigentlich ist es genau das gleiche Verfahren. Beim Drizzeln eines CFA-Bildes bestimmt die CFA-Farbe des aktuellen Tröpfchens, auf welchem Kanal des Ausgabebildes es landet, während beim Drizzeln eines Monobildes alle Tröpfchen auf dem (einzigen) Mono-Ausgabekanal landen. Das Drizzling von CFA-Bildern vermeidet die Artefakte, die bei allen Debayering-Algorithmen auftreten, was zu verbesserten Rauscheigenschaften bei stark gestreckten Bildern führt. Dies ermöglicht eine bessere Rauschunterdrückung und Dekonvolution für gedrizzelte CFA-Sequenzen im Vergleich zu debayerten und registrierten CFA-Bildern und lässt die dunklen Bildanteile besser aussehen.
- **Artefakte vermeiden** Es ist möglich, eine Sequenz mit scale = 1, pixel fraction = 1 zu drizzeln und im Wesentlichen das gleiche Ergebnis zu erzielen wie bei der Anwendung der Registrierung mit einer der interpolierten Methoden. Sie sollten dies in Erwägung ziehen, wenn Sie Interpolationsartefakte bei der standardmäßigen interpolierten Methode sehen (obwohl diese im Allgemeinen durch die Klammerfunktion effektiv unterdrückt werden). Beachten Sie, dass Drizzle selbst verschiedene Artefakte erzeugen kann (siehe die Überschrift "Einige allgemeine Probleme" weiter unten). Diese können jedoch durch die Wahl des Drizzle-Kerns oder durch eine größere Anzahl von Eingabebildern vollständig vermieden werden und werden im Allgemeinen durch das Stacking perfekt gehandhabt.

9.4.2 Grenzen des Drizzelns

- Drizzle ist im Vergleich zur Interpolation etwas langsamer, insbesondere der bevorzugte quadratische Kernel. Wenn Sie ältere oder langsamere Hardware verwenden, sollten Sie die alte Methode bevorzugen.
- Bei der Verwendung von Upsampling erzielt Drizzle seine Auflösungsverbesserungen auf Kosten eines erhöhten Bildrauschens. Daher sollten Sie beim Drizzeln mehr Integrationszeit sammeln als bei einer interpolationsbasierten Upsampling-Methode.
- Das obige Problem gilt *besonders* für CFA-Bilder. Bedenken Sie, dass nur 1 von 2 Pixeln grün und nur 1 von 4 Pixeln rot oder blau sind. Für den roten oder blauen Kanal bedeutet der CFA-Drizzle daher bereits die gleiche Reduzierung der Tröpfchenabdeckung wie ein 2x-Upscale-Drizzle. Wenn Sie zusätzlich hochskalieren, benötigen Sie so viel Tröpfchenabdeckung wie bei einem 4-fach hochskalierten Drizzle! Daher wird im Allgemeinen empfohlen, CFA-Bilder im Maßstab 1 zu drizzlen.

9.4.3 Vergleich

Das folgende Bild zeigt einen Vergleich zwischen Drizzle und der herkömmlichen Hochskalierungsmethode. Bei dem Bild handelt es sich um Ha, das aus einer OSC-Sitzung mit einem Dualband-Filter extrahiert wurde. Links sehen Sie das Ergebnis des alten OSC_Extract_HaOIII-Skripts, das die Ha-Daten, die von den roten Pixeln in der OSC-Bayer-Matrix erfasst werden, als halbgroßes Bild extrahiert und OpenCV-Upscaling mit Lanczos4-Interpolation verwendet, um ein Bild zu erzeugen, das der Größe des OIII-Bildes entspricht.

Rechts sehen Sie das Ergebnis des aktualisierten Skripts (verfügbar über das Repository siril-scripts), das die von den roten Pixeln in der OSC-Bayer-Matrix erfassten Ha-Daten als halbgroßes Bild extrahiert und mit scale = 2,0, pixel fraction = 0,5 drizzelt, um ein Bild zu erzeugen, das der Größe des OIII-Bildes entspricht.

Bei einer Betrachtung im 100 %-Maßstab wird deutlich, dass der gedrizzelte Stack einen Großteil der Auflösung des optischen Systems wiederherstellt, die durch die geringere Anzahl roter Pixel in der Bayer-Matrix zu gering ist: Es sieht viel schärfer aus, und die Zahlen bestätigen dies: Das durchschnittliche fwhm im linken Bild beträgt 3,59, während es im rechten Bild 3,25 ist.

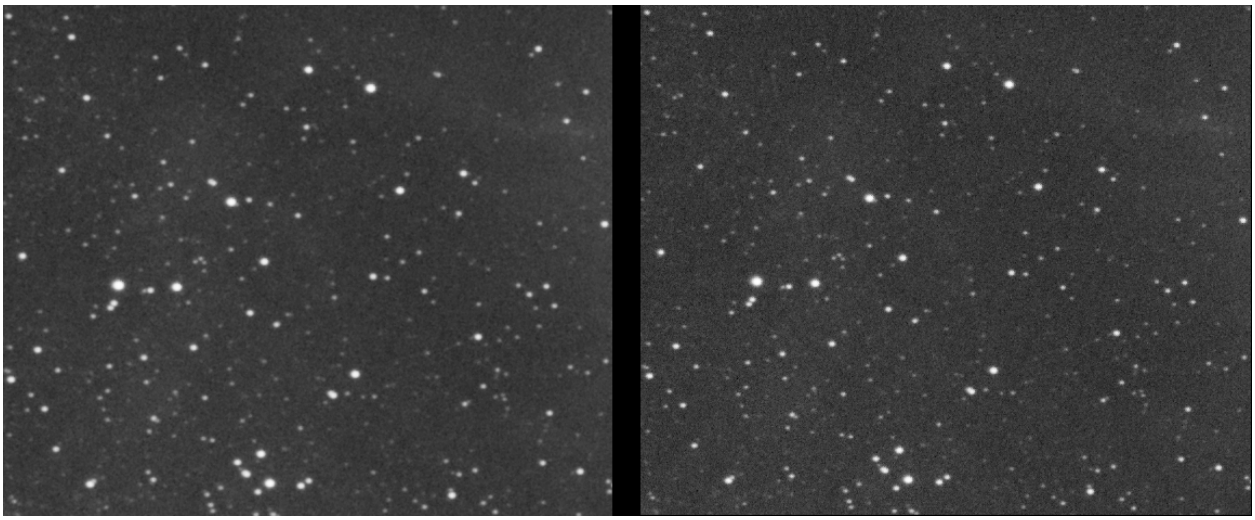


Abb. 34: Vergleich zwischen Hochskalierung durch Interpolation und Drizzle

9.4.4 Arbeitsablauf und Benutzeroberfläche

9.4.5 Arbeitsablauf Mono

Kalibrierung

Bei Monobildern ändert sich auf der Registerkarte "Kalibrierung" nichts. Kalibrieren Sie wie gewohnt.

Registrierung

Drizzeln

Skalierung

Skalierung legt die Skalierung des Drizzle-Ausgangsbildes in Bezug auf das Eingangsbild fest. Ein typischer Drizzle-Maßstab für ein unterabgetastetes Monobild ist 2,0. Das bedeutet, dass die Eingabe auf ein Ausgabepixelraster mit der doppelten Auflösung gedrizzelt wird. (Wenn Ihr Referenz-Eingangsbild 1024 x 512 Pixel groß ist, wäre Ihr Ausgangsbild 2048 x 1024 Pixel groß). Hinweis: Da das Bild denselben Bereich des Himmels darstellt, sind zwar **doppelt so viele** Pixel entlang jeder Achse im Ausgabebild vorhanden, aber jedes Ausgabepixel ist effektiv **halb so breit** und **halb so hoch**.

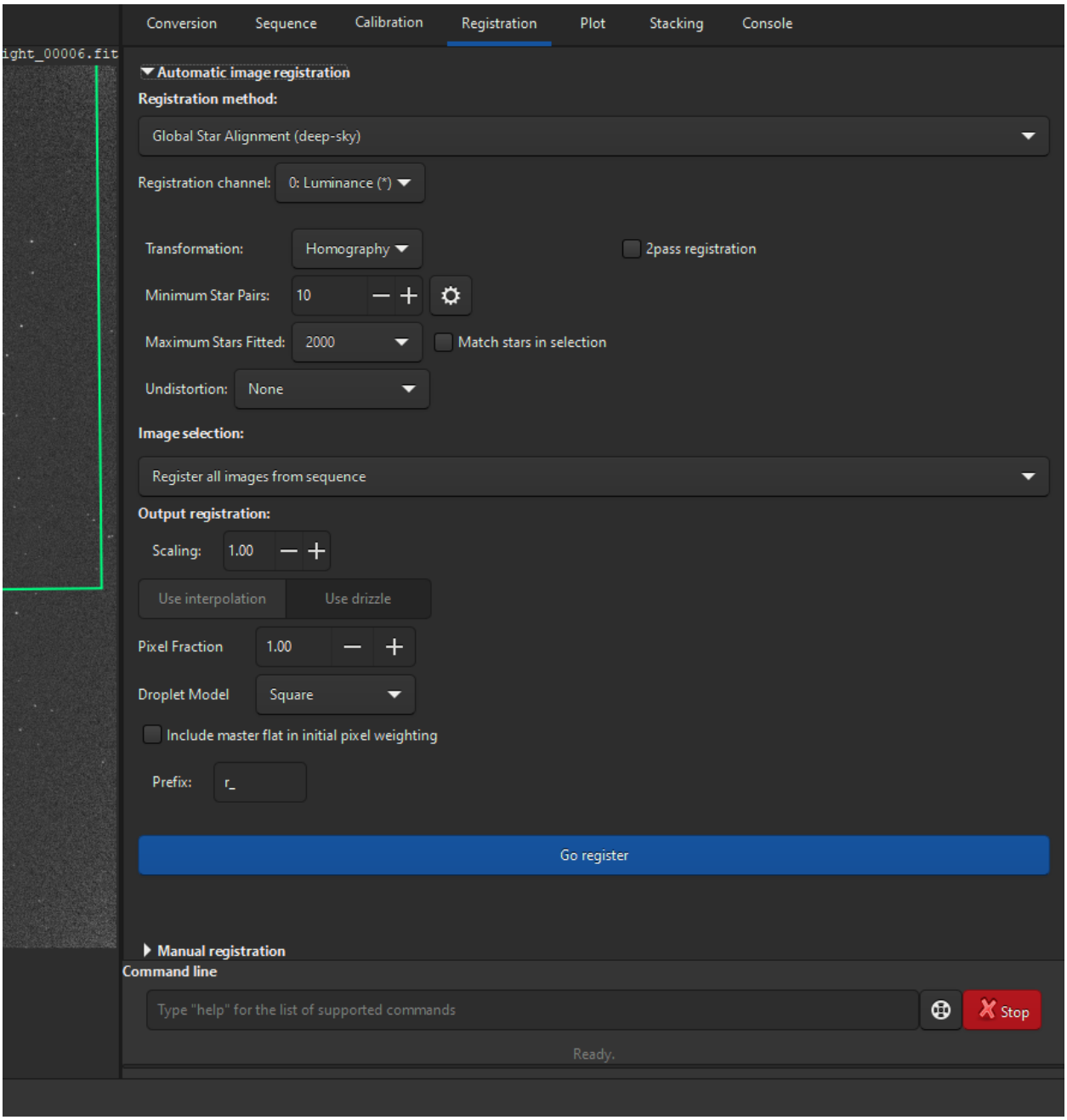


Abb. 35: Registerkarte zeigt de Drizzle-Einstellungen

Tipp

Je größer die Skalierung ist, desto spärlicher ist jedes Drizzled-Ausgangsbild und desto weniger Pixel werden in jedes Ausgangspixel gestackt. Dies führt zu einem stärker verrauschten Bild: **Der Auflösungsgewinn durch Drizzle geht auf Kosten des Rauschens**. Dies muss durch eine längere Gesamtintegrationszeit ausgeglichen werden, als dies ohne Drizzling bei einer höheren Auflösung erforderlich wäre.

Pixelanteil

Pixel Fraction sets the size of the droplet taken from the input grid. Consider a drizzle scale of 2.0: since the output pixels are half as wide and half as tall, that means that in order for each input pixel "droplet" to be the same size as an output pixel it should be shrunk to half the linear dimensions. This is a pixel fraction of 0.5. A good rule of thumb is that the pixel fraction should be roughly the reciprocal of the drizzle scale (with some kernels it helps to set it a little bigger than this, in order to reduce pixels that receive zero input from any drizzled droplets).

Mit dem Pixelanteil kann experimentiert werden: Ein größerer Pixelanteil bedeutet, dass jedes Eingangströpfchen mehr Ausgangspixel beeinflusst. Andererseits bedeutet die Einstellung eines kleineren Pixelanteils, dass jedes Eingangströpfchen weniger Ausgangspixel beeinflusst. Der "Punkt"-Kernel ist ein Sonderfall, bei dem der Pixelanteil gleich Null ist (und bei diesem Kernel hat die Einstellung des Pixelanteils keine Auswirkungen).

Tröpfchenmodell

Die Drizzle-Implementierung von Siril bietet mehrere Tröpfchenmodelle:

- **Quadrat**. Hier wird der Tropfen als quadratischer Tropfen modelliert, der genau mit dem Eingabepixel ausgerichtet ist. Er wird genau auf den Ausgangsreferenzrahmen abgebildet und funktioniert bei jeder Skalierung und jedem Pixelanteil gut. Dies ist der Standardkernel.

  Tipp

  The Square kernel is mathematically flux preserving by construction: each input pixel's flux is distributed into output pixels in exact proportion to area overlap. Total counts are preserved identically. This makes it both robust and ideal for applications where you need strict surface brightness conservation (e.g. diffuse structures or extended sources).

- **Point**. This models the droplet as a point at the center of the input pixel. It is the limit of the square kernel with a very small pixel fraction. This will prodce the lowest correlated noise but is prone to leaving holes unless you have very large numbers of well-dithered frames. If you find holes or artefacts in your final stack, use the Square kernel instead. This kernel is very fast.
- **Turbo**. Hierbei handelt es sich um eine Vereinfachung des Square-Kernels. Er geht davon aus, dass die Rotation zwischen der Eingabe- und der Ausgabereferenz vernachlässigbar ist. Dies führt zu einer wesentlich schnelleren Berechnung, ist jedoch nur eine Annäherung. Es handelt sich um einen „schnellen und ungenauen" Kernel, der ursprünglich für die Erzeugung von Zwischenprodukten innerhalb des HST-Workflows gedacht war, aber aufgrund seiner Geschwindigkeit im Vergleich zu Square auch für das Lucky Imaging von Nutzen sein kann.
- **Gauß**. Hier wird das Tröpfchen als Gaußsche Verteilung modelliert, die auf der Mitte des Eingabepixels zentriert ist und eine Halbwertsbreite (FWHM) von *pixfrac* hat. Die Tröpfchen im Gaußschen Modell sind rund und werden empfohlen, wenn eine verbesserte Erhaltung der PSFs (Sternformen) wichtig ist.

  Tipp

[Avila2015] recommends the Gaussian kernel for applications involving point source photometry. Despite not being mathematically flux preserving, the Gaussian kernel reduces the blockiness typical of square kernels (and avoids the high frequency ringing characteristic of Lanczos kernels, producing smooth, centrally peaked PSFs that are closer to analytical forms used in photometry. While Gaussians don't preserve flux perfectly the trade-off is much better behaved PSFs, which is more important for extracting accurate stellar photometry. It may also be more aesthetically pleasing in a final image.

- **Lanczos2** and **Lanczos3**. These kernels model the droplet as a Lanczos function centered on the center of the input pixel.

  Warnung

  The Lanczos kernels are specially designed for resampling an image from one World Coordinate System (WCS) onto another **at the same scale**. They should only be used with scale == pixfrac == 1.0.

  Lanczos kernels provide a good option for single-image registration at native scale: Lanczos3 produces tigher average FWHM than any other kernel or OpenCV interpolation, but it is significantly slower than other kernels.

Anfängliche Pixelgewichtung

Wenn ein Tropfen auf dem Ausgabepixelraster landet, kann er mehr als ein Ausgabepixel bedecken. Tatsächlich kann ein Ausgabepixel von mehreren Tropfen, nur von einem Bruchteil eines Tropfens oder sogar von gar keinem Tropfen bedeckt sein. Der Beitrag jedes Eingabepixels kann durch den Master-Flat gewichtet werden, sodass Pixel aus Bereichen mit höherem SNR (weniger Vignettierung) stärker gewichtet werden. Sofern Sie keine besonderen Flats haben, macht dies in der Regel keinen praktischen Unterschied.

Um das Master-Flat zu aktivieren, aktivieren Sie das Kontrollkästchen *Master Flat in anfängliche Pixelgewichtung einbeziehen*.

Warnung

Das Masterflat muss in der Registerkarte *Kalibrierung* eingestellt werden!

Lasst uns drizzeln!

Wenn Sie alle Optionen eingestellt haben, klicken Sie erneut auf die Schaltfläche :guilabel:Registrierung ausführen`.

Stacken

Sie können Ihre Drizzle-Sequenz nun wie gewohnt stacken. Beachten Sie, dass bei einigen Kombinationen von Drizzle Skalierung und Tröpfchengröße einige Ausreißerausschlussmethoden besser funktionieren als andere. Insbesondere, wenn Sie eine große Anzahl von "Nulleingabe"- oder Null-Pixeln haben, gibt es weniger Werte, die für die Ausreißererkennung verwendet werden können. MAD kann ein guter Ansatz sein, wenn Ihre übliche Abweisungsmethode nicht funktioniert.

Das folgende GIF zeigt einen Vergleich eines Stacks von 37 Bildern, wobei in einem Fall die Registrierung mittels Interpolation und im anderen Fall die Registrierung mittels Drizzle durchgeführt wurde. Es ist klar erkennbar, dass der Stack, der mit gedrizzelten Daten erstellt wurde, deutlich schärfer ist als der Stack mit Daten, die durch Interpolation registriert wurden.

Abb. 36: Vergleich der Registrierung mit Drizzle und mit Interpolation. Klicken Sie zum Vergrößern der Ansicht.

9.4.6 CFA Arbeitsablauf

CFA Kalibrierung

Für One-Shot-Farbbilder (OSC) deaktivieren Sie das Kontrollkästchen *Vor dem Speichern debayern*. Dies stellt eine Änderung gegenüber früheren Arbeitsabläufen dar, aber für das Drizzling ist es wichtig, dass das CFA-Muster in der Drizzle-Eingabesequenz erhalten bleibt.

CFA Registrierung mit Drizzling

Sie können Drizzle direkt nach einer *globalen Registrierung* verwenden oder durch *Anwenden einer vorhandenen Registrierung* für alle anderen Registrierungsmethoden, die Bilder nicht direkt exportieren.

CFA-Skalierung

Skalierung legt die Skalierung des Drizzle-Ausgangsbildes in Bezug auf das Eingangsbild fest. In OSC-Kamerabildern nimmt jedes Pixel nur eine einzige Farbe auf: Rot, Grün oder Blau. Die Pixel sind mit einer Farbfilteranordnung (CFA) versehen, die bestimmt, welche Pixel auf rote, grüne und blaue Wellenlängen reagieren. Daher sind alle Pixel im Vergleich zu einem Monosensor, bei dem alle Pixel für jedes Licht, das durch den Filter fällt, empfindlich sind, spärlich verteilt. Sowohl bei Bayer-Mustern als auch bei X-Trans-CFAs sind die roten und blauen Pixel in den Eingabebildern besonders spärlich verteilt, so dass bei einer Erhöhung der Drizzle-Skala über 1,0 noch mehr Bilder erforderlich sind, um eine ausreichende Drizzle-Abdeckung zu erzielen und einen akzeptablen Rauschpegel zu erreichen.

Für eine typische OSC-Sensoranwendung, bei der das Seeing gut an die nominale Abtastung des Sensors angepasst ist, wird empfohlen, CFA-Drizzle mit Skalierung (scale) = 1,0 und Pixelanteil (pixfrac) = 1,0 anzuwenden. Dadurch wird die Auflösung in jedem Farbkanal wiederhergestellt (der aufgrund der Abstände der farbigen Pixel in der CFA effektiv unterabgetastet wird) und herkömmliche Debayering-Artefakte vermieden. Wenn Sie das Bild durch Verwendung von scale > 1.0 ebenfalls hochskalieren möchten, sollten Sie sich darüber im Klaren sein, dass die in jedem Kanal verfügbaren Pixel noch spärlicher werden und Sie noch mehr Daten benötigen, um eine angemessene Abdeckung zu gewährleisten und das Rauschen auf ein akzeptables Maß zu beschränken.

Tipp

Für OSC Drizzle beginnen Sie mit Skalierung = Pixelanteil = 1.0.

CFA Pixelanteil

Pixelanteil legt die Größe des aus dem Eingaberaster entnommenen Tröpfchens fest. Hier gelten die gleichen Anmerkungen wie oben für den Mono-Workflow beschrieben.

CFA-Tröpfchenmodell

Für CFA-Drizzle gilt die gleiche Auswahl an Drizzle-Kernen wie für Mono-Drizzle. Beachten Sie, dass die Kernel, die besonders anfällig für die Erzeugung von Null-Pixeln sind, bei der Verwendung für CFA-Drizzle problematisch sein können. Wenn Sie Zehntausende von Einzelbildern wie in einem Planetenvideo haben, kann Turbo gut funktionieren (und wird schnell sein!), aber für Deep-Sky-Sequenzen mit einer geringeren Anzahl von Einzelbildern wird empfohlen, sich an die *quadratischen* oder *Gauß'schen* Kernel zu halten (und bedenken Sie, wie oben erwähnt, dass Gauß nicht flux-erhaltend ist, wenn Sie also etwas mit photometrischen Techniken machen wollen, ist *quadratisch* vorzuziehen).

CFA Anfängliche Pixelgewichtung

Wie beim Mono-Drizzling kann ein Master-Flat angegeben werden. Um die Master-Flat zu aktivieren, aktivieren Sie das Kontrollkästchen *Master-Flat in anfängliche Pixelgewichtung einbeziehen*.

Warnung

Das Masterflat muss in der Registerkarte *Kalibrierung* eingestellt werden!

Auf geht's Bayer Drizzle!

Wenn Sie alle Optionen eingestellt haben, klicken Sie erneut auf die Schaltfläche :guilabel:Registrierung ausführen`.

Stacken Ihrer CFA-Daten

Sie können Ihre Drizzle-Sequenz nun wie gewohnt stacken und dabei die gleichen Anmerkungen zur Ausreißerunterdrückung wie bei Mono-Drizzle beachten (diese können bei CFA-Drizzle deutlicher ausfallen, wenn Sie aufgrund der geringeren Anzahl von Eingangspixeln in jedem Kanal keine ausreichende Abdeckung zur Unterstützung einiger Algorithmen zur Ausreißerunterdrückung haben).

Tipp

Wenn Sie Ihre CFA-Daten drizzeln, um die Auflösung zu erhöhen, kann es sein, dass Sie enttäuscht sind, wenn Sie die Ergebnisse mit gestackten, debayerten Bildern vergleichen. Im Allgemeinen *gibt* es Gewinne, aber sie können marginal sein (z. B. ein paar Prozent Verbesserung bei fwhm) und sind im Allgemeinen nicht annähernd so beeindruckend wie die Auflösungsgewinne, die durch das Drizzling unterabgetasteter Monodaten (Undersampling) erzielt werden können.

Der Grund dafür ist, dass das Debayering bereits einen Teil der verlorenen Auflösung wiederherstellt. Die verschiedenen Debayering-Algorithmen arbeiten unterschiedlich, aber im Allgemeinen beruhen sie alle auf den Grundsätzen der räumlichen und spektralen Korrelation, um einen Teil der in einem Kanal fehlenden Auflösung auf der Grundlage der aus den anderen Kanälen gewonnenen Informationen abzuleiten. [Losson2010]

Der eigentliche Grund für das Drizzeln von CFA-Daten ist, dass das Drizzle-Ergebnis ein viel saubereres Rauschen aufweist. Es sieht weniger "körnig" aus (d. h. es fehlt die Struktur, die im Hintergrund eines typischen debayerten CFA-Stacks zu sehen ist) und lässt sich daher leichter mit Rauschunterdrückungstechniken reduzieren und liefert konsistentere Daten für photometrische Anwendungen wie die Farbkalibrierung. Wenn das Bild stark gestreckt wird, um schwache Merkmale knapp über dem Hintergrund hervorzuheben, sieht der resultierende Hintergrund natürlicher aus.

Vergleich Bayer Drizzle

Die Animation unten zeigt einen Vergleich zwischen CFA-Drizzle mit zwei verschiedenen Pixelanteilen und zwei der klassischen Debayering-Algorithmen.

Abb. 37: Vergleich von CFA Drizzle (hier als Bayer Drizzle bezeichnet) mit klassischen Debayering-Algorithmen

- VNG dient als grundlegende Referenz: Beachten Sie das Farbartefakt um die helleren Sterne.
- RCD ist recht gut für runde Objekte wie Sterne geeignet.
- Bayer Drizzle 1.0 liefert Ergebnisse, die denen von RCD sehr nahe kommen, jedoch mit einem geringeren Rauschen und besseren Hintergrund
- Bayer Drizzle 0,5 bietet eine bessere Auflösung auf Kosten von mehr Rauschen. Der Kompromiss, den der Pixelanteil zwischen Auflösung und Rauschen darstellt, ist offensichtlich. Bei einem kleineren Pixelanteil benötigt CFA-Drizzle mehr Daten, um das gleiche Signal-Rauschverhältnis zu erzielen.

9.4.7 Einige häufige Probleme

> **Tipp**
>
> **KEINE PANIK** - die folgenden Ergebnisse sehen vielleicht etwas seltsam aus, wenn Sie ein einzelnes Drizzle-Sub ansehen, aber sie sind keine Fehler - der Algorithmus funktioniert wie vorgesehen. In den meisten Fällen lösen sie sich beim Stacken von selbst auf, in den übrigen Fällen können sie durch Ändern der Drizzle-Parameter oder durch Einfügen weiterer Datenbilder behoben werden.

Moiré Muster

Aufgrund der Beschaffenheit des Drizzle-Algorithmus erhalten einige Ausgabepixel bei der Hochskalierung möglicherweise keine Daten. Diese werden als "Nullpixel" bezeichnet und haben einen Nullwert. Einige Kernel kompensieren dies, indem sie den Pixelanteil begrenzen, so dass alle Ausgangspixel einen gewissen Input erhalten, andere hingegen nicht.

Ausgabepixel, die keinen Eingabedaten erhalten, sind schwarz: Da sie typischerweise in Mustern auftreten, die auf der Geometrie der Transformation aus dem Eingabebild basieren, sehen sie typischerweise wie Moiré-Muster aus, wie unten gezeigt:

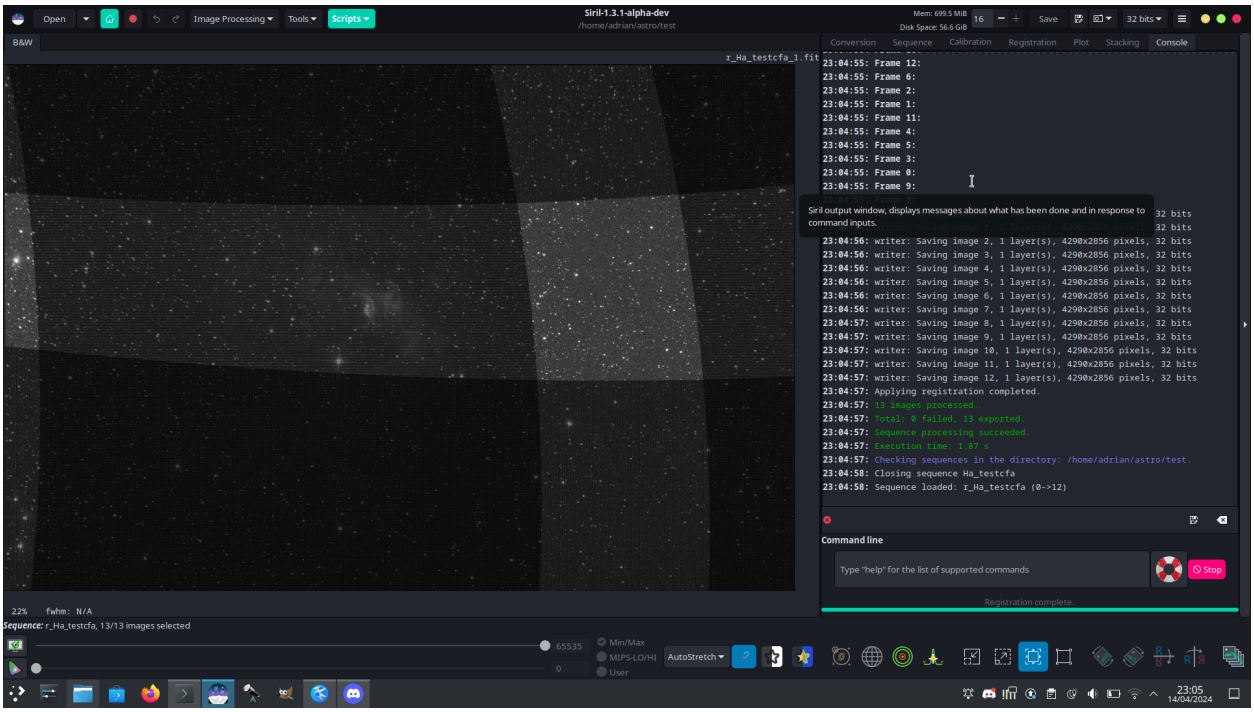


Abb. 38: Darstellung von Mustern, die sich aus Null-Pixeln in einem gedrizzelten Bild ergeben

Machen Sie sich darüber keine Sorgen! Siril ignoriert Pixel, die beim Stacking genau 0 sind. Solange Sie also genügend Eingabeframes haben und die Dither-Positionen angemessen gestreut sind, werden alle Pixel von genügend Pixeln abgedeckt und der Ausgabestapel ist in Ordnung. Wenn Sie jedoch mit einer geringeren Anzahl von Eingabebildern stacken und dies zu Problemen führt, sollten Sie einen anderen Drizzle-Kernel verwenden. Hier sehen Sie genau das gleiche Bild, das mit dem gleichen Maßstab und Pixelanteil, aber mit dem quadratischen Kernel anstelle des Turbo-Kernels gedrizzelt wurde. Das Ergebnis ist anders, und die Muster sind nicht mehr erkennbar.

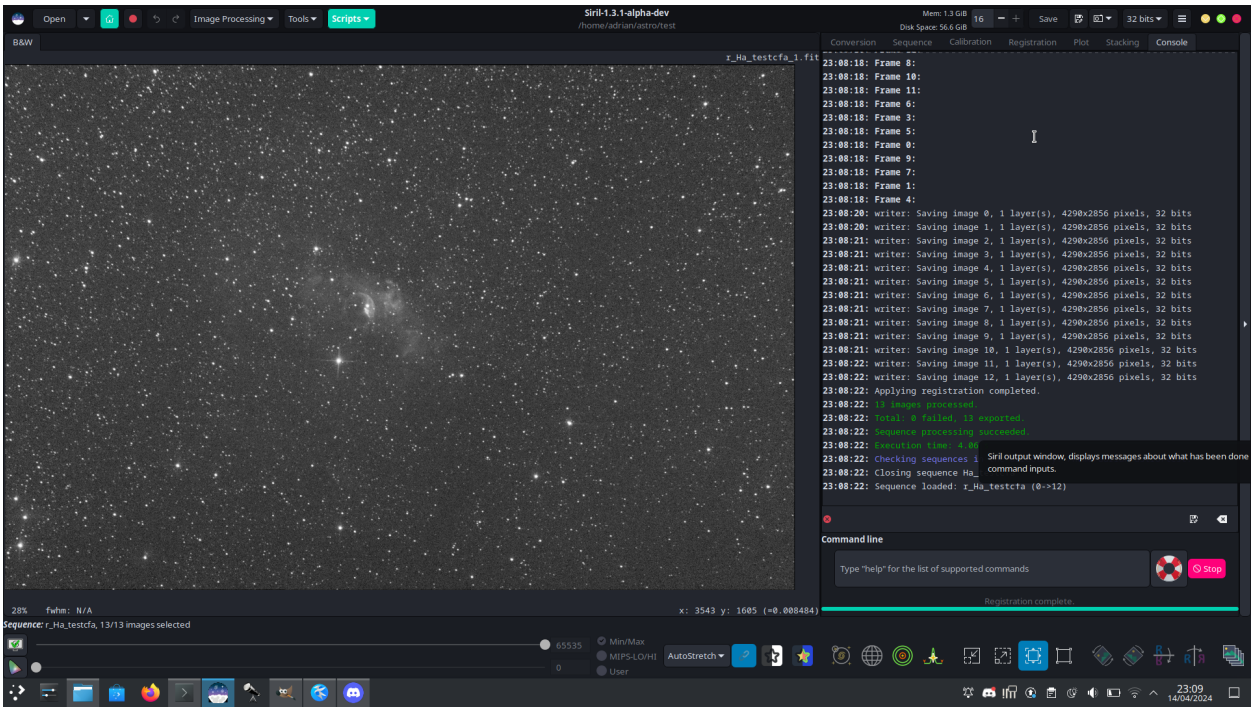


Abb. 39: Die Verwendung eines anderen Drizzle-Kerns kann die fleckigen Muster von Null-Pixeln eliminieren

Fleckige Stapel

Ein Problem, das beim Stacken von gedrizzelten Daten auftreten kann, wenn zu viele Null-Pixel vorhanden sind, ist, dass das Endergebnis seltsam fleckig aussehen kann:

Dies tritt typischerweise bei Punkt-, Turbo- oder Lanczos-Kerneln auf. Sie können es beheben, indem Sie den quadratischen oder den Gauß-Kernel verwenden oder mehr Eingabebilder nutzen.

Befehle

Siril Kommandozeile

```
seqapplyreg sequencename [-prefix=] [-scale=] [-layer=] [-framing=]
seqapplyreg sequencename ... [-interp=] [-noclamp]
seqapplyreg sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
seqapplyreg sequencename ... [-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-
↪filter-round=value[%|k]] [-filter-bkg=value[%|k]] [-filter-nbstars=value[%|k]] [-
↪filter-quality=value[%|k]] [-filter-incl[uded]]
```

Wendet geometrische Transformationen auf Bilder der im Argument angegebenen Sequenz an, so dass sie dem Referenzbild überlagert werden können, wobei zuvor berechnete Registrierungsdaten verwendet werden (siehe REGISTER).

Der Name der Ausgabesequenz beginnt mit dem Präfix **"r_"**, sofern mit der Option **-prefix=** nichts anderes angegeben wurde.

Bei RGB-Bildern wird die Registrierung auf der ersten Ebene durchgeführt, für die Daten vorhanden sind, es sei denn, die Option **-layer=** (0, 1 oder 2 für R, G bzw. B) wurde angegeben.

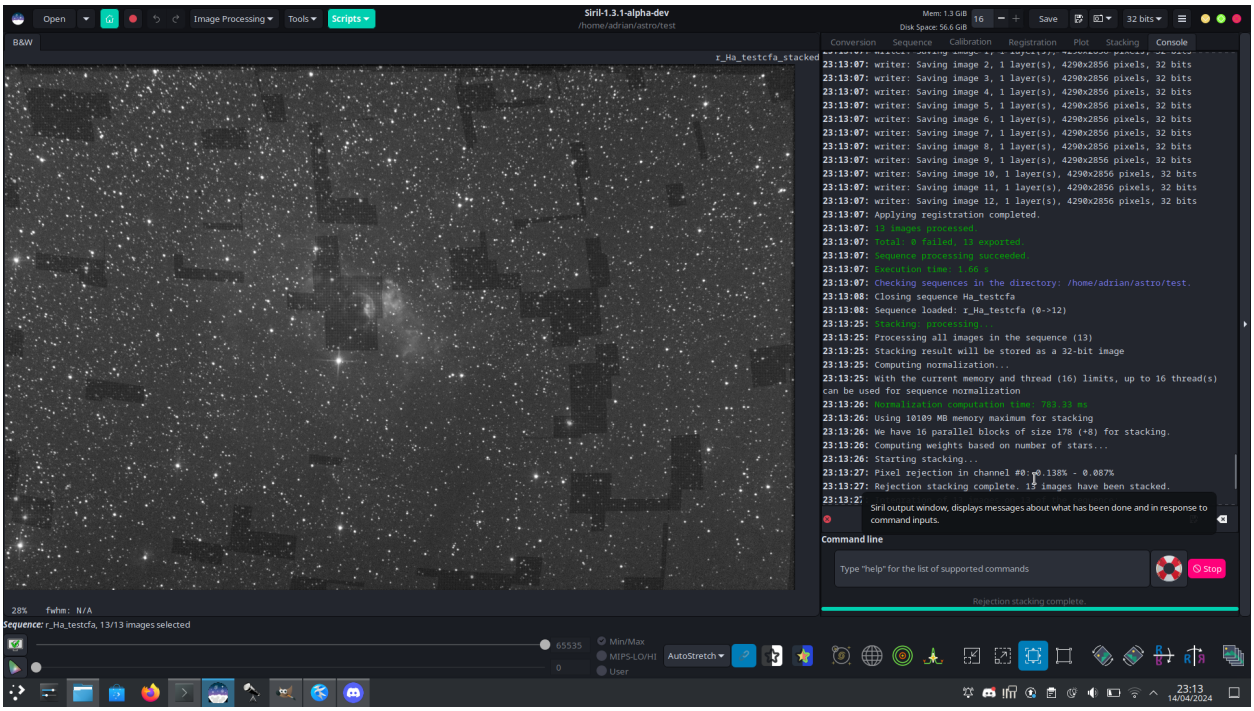


Abb. 40: Typisches uneinheitliches Erscheinungsbild eines Stacks von gedrizzelten Daten mit zu vielen Null-Pixeln / zu wenig Frames

Die ausgegebenen Bilder können durch Übergabe eines Wertes für das Argument **-scale=** (Skalierung, zwischen 0,1 und 3) vergrößert oder verkleinert werden.

Die automatische Beschneidung der Ausgabesequenz kann mit dem Schlüsselwort **-framing=**, gefolgt von einer der Methoden aus der Liste { current | min | max | cog } festgelegt werden:

-framing=max (Begrenzungsrahmen) projiziert jedes Bild und berechnet seine Verschiebung in Bezug auf das Referenzbild. Die resultierende Sequenz kann dann mit der Option **-maximize** des STACK-Befehls gestackt werden, wodurch das vollständige Bild erstellt wird, das alle Bilder der Sequenz umfasst.

-framing=min (gemeinsamer Bereich) beschneidet jedes Bild auf den Bereich, den es mit allen Bildern der Sequenz gemeinsam hat.

-framing=cog bestimmt die beste Position für den Bildausschnitt als Schwerpunkt (cog) aller Bilder.

Optionen zur Bildtransformation:

Standardmäßig werden Transformationen angewendet, um die Bilder mittels Interpolation zu registrieren.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden.

Optionen zum Drizzeln:

Andernfalls können die Bilder mit dem HST-Drizzle-Algorithmus exportiert werden, indem das Argument **-drizzle** übergeben wird, das die folgenden zusätzlichen Optionen annehmen kann:

-pixfrac= legt den Pixelanteil fest (Standard = 1,0).

Das Argument **-kernel=** legt den Drizzle-Kernel fest und muss von einem der folgenden Argumente gefolgt werden: **point**, **turbo**, **square**, **gaussian**, **lanczos2** oder **lanczos3**. Der Standardwert ist **square**.
Das Argument **-flat=** gibt ein Master-Flat an, um die gedrizzelten Eingabepixel zu gewichten (Standard ist kein Flat).

Bilder herausfiltern:
Die zu registrierenden Bilder können auf der Grundlage einiger Filter ausgewählt werden (z. B. derjenigen mit der besten FWHM), mit einigen der **-filter-*** Optionen.

Links: *register*, *stack*

Beachten Sie, dass die Einführung von echtem Drizzlen aus Gründen der Übersichtlichkeit einige Änderungen an vorhandenen Befehlsargumenten erforderlich gemacht hat.

register und ref:*seqapplyreg <seqapplyreg>* haben ein neues Argument **-drizzle**, das, zusammen mit einigen verwandten Argumenten, "true drizzle" aktiviert.

Quellenverzeichnis

9.5 Stacken

Der letzte Schritt der Vorverarbeitung mit Siril ist das Stacken der Bilder. Stacking ist eine Technik, die in der Astrofotografie verwendet wird, um die Qualität und Detailgenauigkeit eines Bildes zu verbessern, indem mehrere Fotos zu einem einzigen zusammengesetzten Bild kombiniert werden. Bei diesem Verfahren werden mehrere Bilder desselben Objekts aufgenommen und anschließend ausgerichtet und gemittelt, um das Rauschen zu reduzieren und das Signal-Rausch-Verhältnis zu erhöhen. Das Ergebnis ist ein endgültiges Bild, das weniger Rauschen, mehr Details und einen größeren Dynamikbereich aufweist als eine Einzelbelichtung.

9.5.1 Stackingmethoden

Summenstacking

Dies ist der einfachste Algorithmus: Jedes Pixel im Stack wird summiert. Der Anstieg des Signal-Rausch-Verhältnisses (SNR) ist proportional zu $\sqrt{N}$‘, wobei N die Anzahl der Bilder ist. Aufgrund der fehlenden Normalisierung und Ausreißerbehandlung sollte diese Methode nur für die Verarbeitung von Planetenbildern verwendet werden.

Bei 8- oder 16-Bit-Eingabebildern pro Kanal wird die Summe in einer 64-Bit-Ganzzahl gebildet, bevor sie auf den maximalen Pixelwert normalisiert und als vorzeichenlose 16-Bit-Ganzzahl oder 32-Bit-Fließkommazahl gespeichert wird.

Diese Stackingmethode sollte für 8-Bit-Eingabebilder verwendet werden, da sie die Dynamik der Bilder während des Stackings erhöht und die Details erkennbar macht. Das Stacken mit einer Mittelwert- oder Median-Methode würde bei einer solchen Sequenz nur das Rauschen verringern, aber nicht die Dynamik des Bildes verbessern, das Ergebnis wäre immer noch 8 Bit tief.

Durchschnittswert-Stacking mit Ausschluss

Bei dieser Stapelmethode wird ein Mittelwert der Pixel in einem Stapel berechnet, nachdem abweichende Pixel ausgeschlossen wurden und eine optionale Normalisierung der Bilder gegenüber dem Referenzbild vorgenommen wurde. Wie beim Summenstacking ist die Verbesserung des SNR proportional zu $\sqrt{N}$. Es gibt verschiedene Möglichkeiten, die Bilder zu normalisieren und abweichende Pixel zu erkennen und zu ersetzen bzw. auszuschließen, die im Folgenden erläutert werden.

Warnung

Einige Betriebssysteme begrenzen die Anzahl der Bilder, die gleichzeitig geöffnet werden können, was für Median- oder Mean-Stacking-Methoden erforderlich ist. Unter Windows liegt die Grenze bei 2048 Bildern. Wenn Sie sehr viele Bilder haben, sollten Sie eine andere Art von Sequenz verwenden, die *hier* beschrieben wird.

Abb. 41: Stapeln mit Ausschluss (Rejection)

Pixelausschlussmethoden

- **Perzentil Clipping**: Dies ist ein einstufiger Ablehnungsalgorithmus, der ideal für kleine Datensätze (bis zu 6 Bilder) ist.
- **Sigma Clipping**: Hierbei handelt es sich um einen iterativen Algorithmus, der Pixel ausschließt, deren Abstand zum Median am weitesten von zwei vorgegebenen Werten in Sigma-Einheiten (σ low, σ high) entfernt ist.
- **MAD-Clipping**: Hierbei handelt es sich um einen iterativen Algorithmus, der wie Sigma Clipping funktioniert, mit dem Unterschied, dass als Abschätzung die absolute Medianabweichung (MAD) verwendet wird. Dieser Algorithmus wird im Allgemeinen für die Verarbeitung verrauschter Infrarotbilder verwendet.
- **Median-Sigma-Clipping**: Dies ist derselbe Algorithmus wie Sigma Clipping, mit dem Unterschied, dass die abgelehnten Pixel durch den Medianwert des Stacks ersetzt werden.
- **Winsorized Sigma Clipping**: Diese Methode ist der Methode Sigma Clipping sehr ähnlich, nur dass sie robuster bei der Erkennung von Ausreißern sein soll, siehe Hubers Arbeit [Peter2009].
- **Generalisierter extrem studentisierter Abweichungstest (Generalized Extreme Studentized Deviate Test)** [Rosner1983]: Hierbei handelt es sich um eine Verallgemeinerung des Grubbs-Tests, der dazu dient, einen oder mehrere Ausreißer in einem univarianten Datensatz zu erkennen, der einer annähernden Normalverteilung folgt. Dieser Algorithmus zeigt hervorragende Leistungen bei großen Datensätzen mit mehr als 50 Bildern.
- **Linear Fit Clipping** [ConejeroPI]: Er passt die beste gerade Linie ($y = ax + b$) des Pixelstacks an und verwirft Ausreißer. Dieser Algorithmus funktioniert sehr gut bei großen Stapeln und Bildern, die Himmelsgradienten mit unterschiedlichen räumlichen Verteilungen und Ausrichtungen enthalten.

Beachten Sie, dass Sie bei gedrizzelten CFA-Bildern mit einigen Methoden Schwierigkeiten haben können, eine zufriedenstellende Unterdrückung (z.B. von Satellitenspuren) im roten und blauen Kanal zu erreichen. In diesen Fällen scheint die MAD-Clipping-Methode am effektivsten zu sein.

Pixelausschlusskarten

Die Option *Ausschlusskarte (Rejection Map) erstellen* berechnet und erstellt während des Stackings Rejection Maps. Dies sind Bilder, die zeigen, wie viele Bilder für jedes Pixel des Ergebnisbildes zurückgewiesen wurden, geteilt durch die Anzahl der gestapelten Bilder. Wenn *L+H zusammenführen* angekreuzt ist, erstellt Siril nur eine Rejection Map, die die Summe der Low und High Maps ist.

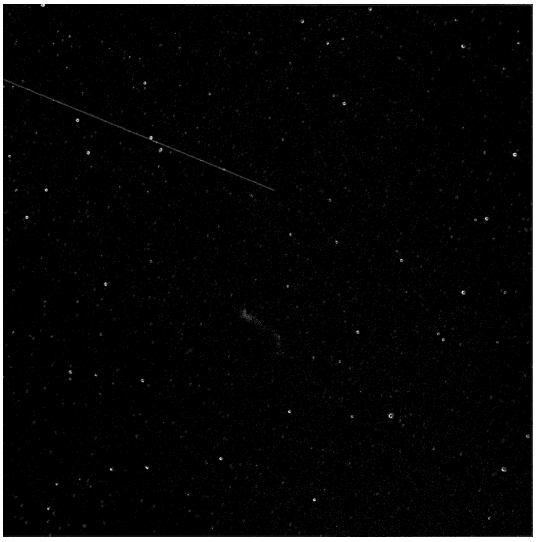


Abb. 42: Beispiel einer Ausschlusskarte (L+H). Wir können sehr deutlich die Spur eines entfernten Satelliten erkennen.

Bilder filtern/gewichten

Die Gewichtung ermöglicht es, jedem Bild ein statistisches Gewicht zuzuweisen. Auf diese Weise tragen die Bilder, die als die besten gelten, mehr bei als die, die als die schlechtesten gelten. Es gibt vier Methoden der Gewichtung:

- **Anzahl Sterne** gewichtet einzelne Bilder auf der Basis der Anzahl berechneten Sterne während der Registrierung.
- **Gewichtetes FWHM** gewichtet einzelne Frames basierend auf dem wFWHM-Wert, der während des Registrierungsschritts berechnet wurde. Dies ist ein FWHM, gewichtet mit der Anzahl der Sterne im Bild. Bei gleichem FWHM-Wert hat ein Bild mit mehr Sternen ein besseres wFWHM als ein Bild mit weniger Sternen.
- **Rauschen** gewichtet die Bilder auf der Basis des Hintergrundrauschens.
- **Anzahl der Bilder** gewichtet einzelne Bilder auf der Basis der Integrationszeit.

Bildzusammenfügung

Dieses Untermenü ist seit Siril 1.3 verfügbar und ermöglicht ein gleichmäßigeres Zusammenfügen von Bildern mit geringen Überlappungen. Es ist der erste Schritt zum Mosaik-Zusammenfügen, das wir in den nächsten Versionen weiter ausbauen möchten.

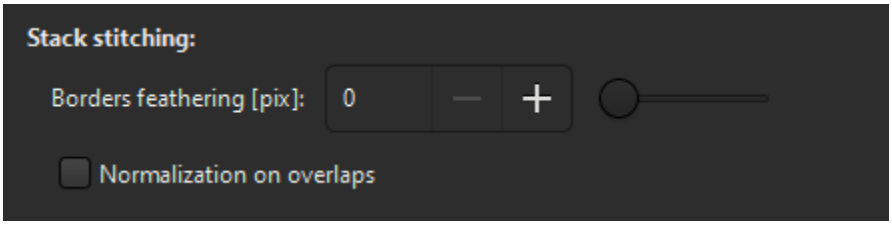


Abb. 43: Stacking stiching panel

- *Ränder überblenden* wendet beim Stacking eine Überblendungsmaske auf jedes Bild an. Die Glättung wird über einen Abstand von X Pixeln angewendet, wobei der Wert zwischen X Pixeln liegt, den Sie mit dieser Schnittstelle festlegen. Die Masken werden im Unterordner `./cache` zwischengespeichert.

Abb. 44: Randverläufe auf einem 3x2-Mosaik (Negativ) angewendet. Bilder mit freundlicher Genehmigung von G. Attard

- *Normalization on overlaps* will compute the normalization factors on the images overlaps instead of whole images. This is useful when images have little overlap and when the nature of the data in each tile is very different accross all images (some images with mostly nebulosity and others with mainly black sky). As it is longer to compute than regular normalization, you should try without ticking this option first. This feature is intended for stitching stacked mosaic tiles so it will throw a warning if you try to compute it for a long sequence (*i.e.* for subs acquired with a smart telescope in mosaic mode). It is only available if *maximize framing* is enabled.

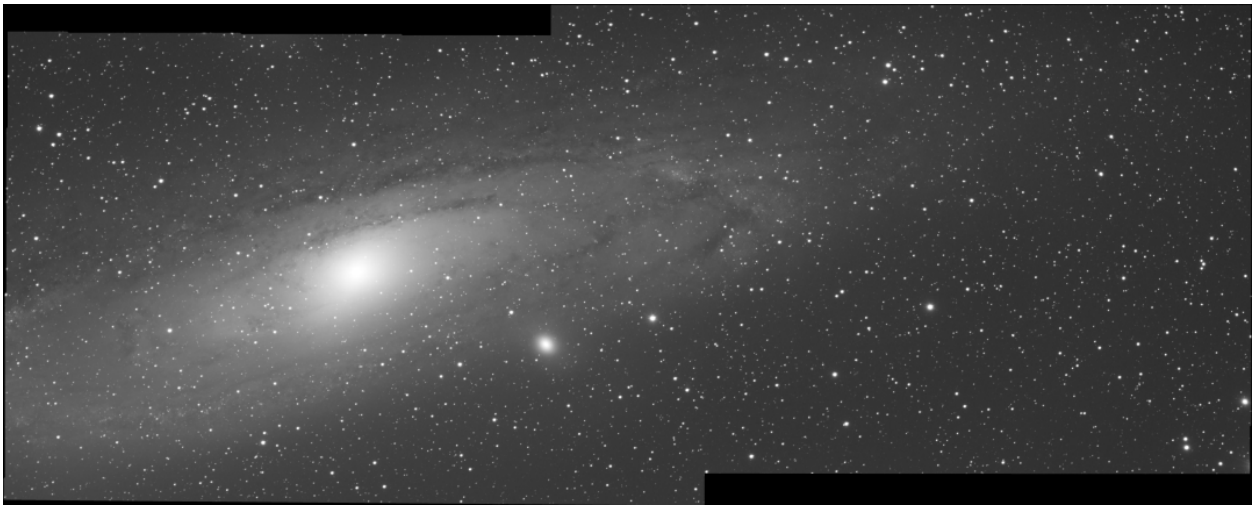


Abb. 45: Globale Normalisierung vs. Normalisierung an Überlappungen auf einem 2x1 Mosaik. Bilder mit freundlicher Genehmigung von D. Huber

Theorie

Die Normalisierung an Überlappungen wurde von einem Algorithmus für Gain-Kompensierung von [Brown2007] abgeleitet. Alle Modifizierungen werden unter der Dokumentation der Mitwirkenden beschrieben.

Median-Stacking

Diese Methode wird meist für Dark/Flat/Bias-Stacking verwendet. Der Medianwert der Pixel im Stapel wird für jedes Pixel berechnet.

Der Gewinn des Signal-Rauschverhältnisses ist proportional zu $0.8\sqrt{N}$ und ist damit geringer als beim Durchschnitts-Stacking, welches generell zu bevorzugen ist.

Pixel Maximum Stacking

Dieser Algorithmus wird hauptsächlich für die Erstellung von Sternenspurenbildern (Star Trail Images) mit langer Belichtungszeit verwendet. Die Pixel des Bildes werden durch Pixel an denselben Koordinaten ersetzt, wenn die Intensität größer ist.

Pixel Minimum Stacking

Dieser Algorithmus wird hauptsächlich zum Beschneiden von Sequenzen verwendet, indem schwarze Ränder entfernt werden. Die Pixel des Bildes werden durch Pixel an denselben Koordinaten ersetzt, wenn die Intensität geringer ist.

9.5.2 Methoden zur Eingangsnormalisierung

Durch die Normalisierung werden die Helligkeitswerte der einzelnen Bilder an das Referenzbild angepasst. Dies ist besonders nützlich für das Durchschnittswert-Stacking mit Ausschluss, da die Ablehnung von Pixeln, wenn die Bilder Helligkeitsunterschiede aufweisen, nicht sehr sinnvoll ist. Diese Abweichungen können durch leichten Nebel/Wolken, durch den Mond oder die Lichter der Stadt verursachte Lichtgradienten, Temperaturschwankungen des Sensors usw. verursacht werden.

Dadurch wird das Signal-Rausch-Verhältnis tendenziell verbessert, weshalb diese Option bei der additiven Normalisierung standardmäßig verwendet wird.

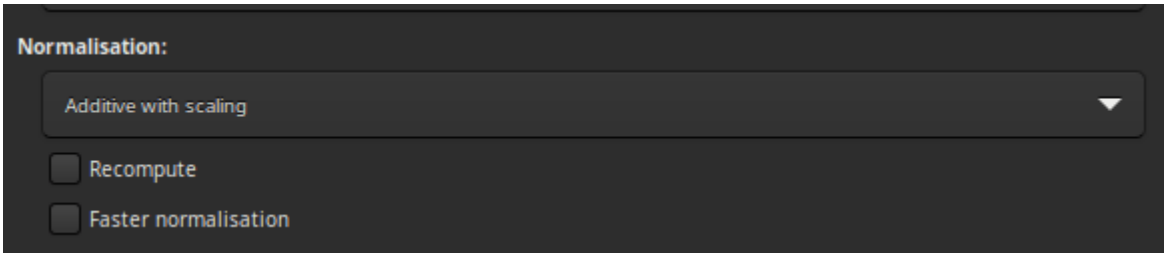


Abb. 46: Normalisierungsfeld für Stapeleingabe

Wenn einer dieser 5 Punkte ausgewählt wird, wird vor dem Stacking ein Normalisierungsprozess auf alle Eingabebilder angewendet.

- Die Normalisierung entspricht dem mittleren Hintergrund aller Eingangsbilder und wird dann durch Multiplikation oder Addition verarbeitet. Beachten Sie, dass beide Verfahren im Allgemeinen zu ähnlichen Ergebnissen führen, aber die multiplikative Normalisierung wird für Bilder bevorzugt, die zur Multiplikation oder Division als Flatfield verwendet werden sollen.
- Die Skalierung entspricht der Streuung durch Gewichtung aller Eingangsbilder. Dadurch wird das Signal-Rausch-Verhältnis tendenziell verbessert, weshalb diese Option standardmäßig mit der additiven Normalisierung verwendet wird.

Normalisierung	Operation	Anwendungsfall
Keine	Es wird keine Normalisierung angewendet.	Dark/Bias-Frames
Additiv	Mittelere Hintergrundwerte werden durch die Anwendung additiver Operationen angeglichen.	
Multiplikativ	Eine Division wird benutzt um die mittleren Hintergrundwerte anzugleichen.	Flat-Frames
Additiv + Skalierung.	In Kombination mit dem additiven Hintergrund durch additives Matching werden die Bilder skaliert, um eine Dispersionsanpassung zu erreichen.	Light-Frames
Multiplikativ + Skalierung.	In Kombination mit der Hintergrundanpassung durch Division werden die Bilder skaliert, um eine Dispersionsanpassung zu erreichen.	

Bemerkung

Die Bias- und Dark-Master sollten nicht mit Normalisierung verarbeitet werden. Bei Flat-Field-Bildern muss jedoch eine multiplikative Normalisierung durchgeführt werden.

Beachten Sie, dass beide Verfahren im Allgemeinen zu ähnlichen Ergebnissen führen, aber die multiplikative Normalisierung wird für Bilder bevorzugt, die zur Multiplikation oder Division als Flat verwendet werden sollen.

Da der Schritt der Normalisierungsberechnung in der Regel langwierig ist, da er die Ermittlung aller statistischen Daten des Bildes erfordert, werden die Ergebnisse in der Datei `seq` gespeichert. Auf diese Weise kann der Benutzer, wenn er ein weiteres Stacking durchführen möchte, indem er die Ablehnungsparameter ändert, die Berechnung schneller durchführen. Die Option *Recompute* erlaubt es, die Neuberechnung der Normalisierung zu erzwingen.

Standardmäßig verwendet Siril zur Berechnung der Normalisierung IKSS-Schätzer für Ort und Maßstab. Bei langen Sequenzen kann die Berechnung dieser Schätzer recht aufwendig sein. In solchen Fällen können Sie mit der Option *Schnellere Normalisierung* schnellere Schätzer (basierend auf dem Median und der absoluten Medianabweichung) wählen. Diese sind zwar weniger widerstandsfähig gegen Ausreißer in den einzelnen Bildern, können aber dennoch ein zufriedenstellendes Ergebnis liefern, wenn man sie mit einer Berechnung ohne Normalisierung vergleicht.

9.5.3 Bilder ausschließen

Es ist auch möglich, eine bestimmte Anzahl von Bildern auszuschließen, um nur die besten Bilder auszuwählen. Dies kann für Lucky-DSO-Techniken sehr nützlich sein, wenn die Anzahl der Bilder in einer Sequenz sehr hoch ist. Man kann zwischen % und k-σ wählen, um entweder einen bestimmten Prozentsatz der Bilder zu behalten oder den zulässigen Schwellenwert mit Hilfe von k-σ zu berechnen.

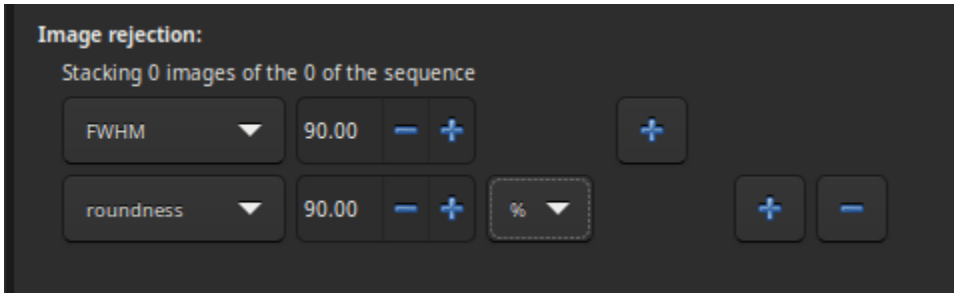


Abb. 47: Stapeln mit Ausschluss (Rejection)

Es sind mehrere Kriterien verfügbar:

- **Alle**: Alle Bilder der Sequenz werden im Stapel verwendet.
- **Ausgewählter Bereich**: nur Bilder verwenden, die in der Sequenz nicht abgewählt wurden.
- **FWHM**: Bilder mit der besten berechneten FWHM (nur sternbasierte Registrierung).
- **gewichtete FWHM**: Dies ist eine Verbesserung der einfachen FWHM. Sie ermöglicht es, viel mehr fehlerhafte Bilder auszuschließen, indem sie die Anzahl der erkannten Sterne im Vergleich zum Referenzbild verwendet (Nur bei sternbasierter Registrierung).
- **Rundheit**: Bilder mit der besten Sternrundheit (nur sternbasierte Registrierung).
- **Hintergrund**: Bilder mit den niedrigsten Hintergrundwerten (nur sternbasierte Registrierung).
- **nb Sterne**: Bilder mit der höchsten Anzahl von erkannten Sternen (nur sternbasierte Registrierung).
- **Qualität**: Bilder mit der besten Qualität (planetarische DFT- oder Kombat-Registrierungen).

9.5.4 Stacking Ergebnis

- Wenn *Ausgabe Normalisierung* angekreuzt ist, wird das endgültige Bild im Bereich [0, 1] normalisiert, wenn Sie im 32-Bit-Format arbeiten, oder sonst im Bereich [0, 65535].

> **Warnung**
>
> Diese Option sollte beim Master-Stacking nicht aktiviert werden.

- Wenn *RGB-Gewichtung* aktiviert ist, werden die Kanäle im endgültigen Bild ausgeglichen (nur Farbbilder).
- Falls *Maximale Rahmung* eingeschaltet ist, wird das resultierende Bild alle Bilder enthält. Diese Option ist immer eingeschaltet, wenn die Bilder unterschiedlich groß sind.
- Wenn :guilabel:`Interpolation Hochskalierung x2`ist ausgewählt, werden Bilder während dem Stacken hochskaliert. Diese Option ist jedoch nicht immer verfügbar.
- Das Auswählen von :guilabel:`Erzwinge 32b`erzwingt, dass das gestackte Bild als Float-Bild abgespeichert wird, auch wenn die Bittiefe in den Einstellungen anders gesetzt ist.
- Das Stacking-Ergebnis wird unter dem im Textfeld angegebenen Namen gespeichert. Es ist möglich, *Pfadparsing* zu verwenden, um den Dateinamen zu erstellen. Ein Klick auf die Schaltfläche *Überschreiben* erlaubt es der neu erstellten Datei, die alte zu überschreiben, falls sie existiert. Wenn letzteres nicht angekreuzt ist, aber bereits ein Bild mit demselben Namen existiert, wird keine neue Datei erstellt.

9.5.5 Quellenverzeichnis

KAPITEL 10

In Bearbeitung

Dieser Abschnitt führt Sie durch die verschiedenen Bearbeitungsschritte Ihrer Bilder. Das Dropdown-Menü ist über die Kopfzeile über die Schaltfläche *Bildbearbeitung* erreichbar. Die Werkzeuge sind in den Menüs und Untermenüs sowie auch in dieser Dokumentation nach Themen gruppiert.

10.1 Stretching der Bilder

Die Bilder werden als Pixelwerte gespeichert, die von der Kamera kommen und einer quasi-linearen Gesetzmäßigkeit folgen. Das bedeutet, dass für Bereiche des Himmels, die keine sichtbaren Merkmale aufweisen, der Pixelwert nahe bei Null liegt, während er für helle Objekte wie Sterne je nach Belichtung und Verstärkung nahe bei einem Maximalwert liegt. Wenn ein Nebel eine Oberflächenhelligkeit hat, die halb so groß ist wie die eines Sterns, hat er Pixelwerte, die halb so groß sind wie die des Sterns, und so weiter. Dies ist der so genannte lineare Pixelmodus.

Das menschliche Auge sieht Photonen nicht auf die gleiche Weise. Es verstärkt dunkle Bereiche, so dass ein Objekt, das vielleicht ein Zehntel so hell ist wie ein anderes, nur halb so hell erscheint. In der Astronomie werden Bilder in der Regel mit einer ähnlichen Pixelwert-Skalierung angezeigt (siehe Anzeigemodi in der grafischen Benutzeroberfläche).

Es handelt sich jedoch nur um einen Anzeigetrick, bei dem eine Bildschirmübertragungsfunktion verwendet wird, um die Pixelwerte des unveränderten Bildes in besser aussehende Bilder umzuwandeln.

Beim Stretching geht es darum, etwas Ähnliches zu tun, indem die Pixelwerte von Bildern geändert werden, anstatt nur ihre Darstellung zu verändern. Siril verfügt über drei Hauptwerkzeuge, um dies zu erreichen.

10.1.1 Asinh-Transformation

Die asinh-Transformation (inverser hyperbolischer Sinus) verändert die Pixelwerte des Bildes in ähnlicher Weise wie die asinh-Pixelskalierungsfunktion, die durch die Schieberegler für niedrige und hohe Werte parametrisiert wird. Hier sind die Parameter der Streckungsfaktor und der Schwarzpunktwert.

Bei monochromen Bildern werden die Pixelwerte nach der folgenden Funktion geändert:

$$\text{Pixel} = \frac{(\text{Original} - \text{Schwarzpunkt}) \times \text{Asinh}(\text{Original} \times \text{Streckung})}{\text{Original} \times \text{Asinh}(\text{Streckung})}$$

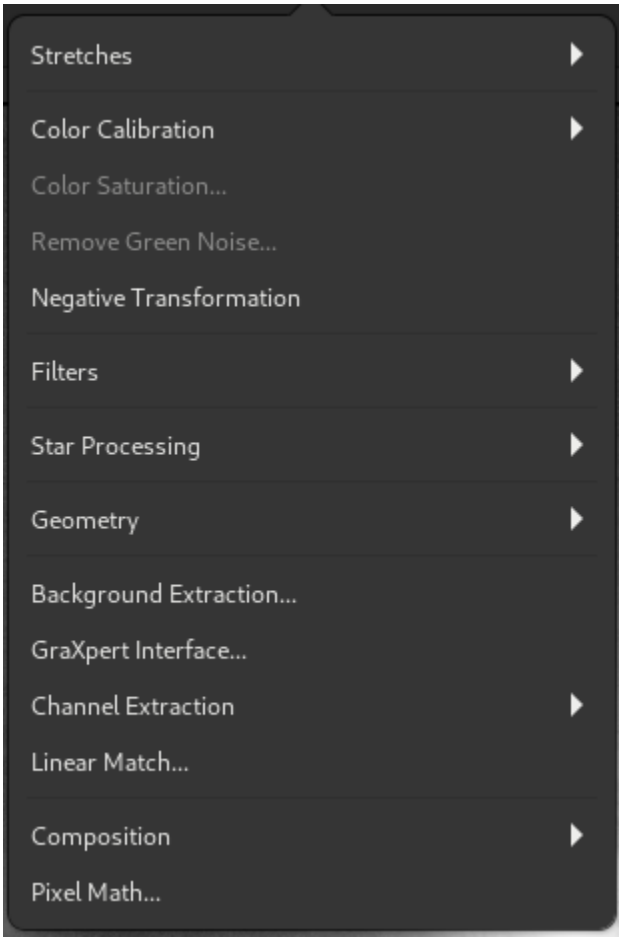


Abb. 1: Bildverarbeitungsmenü

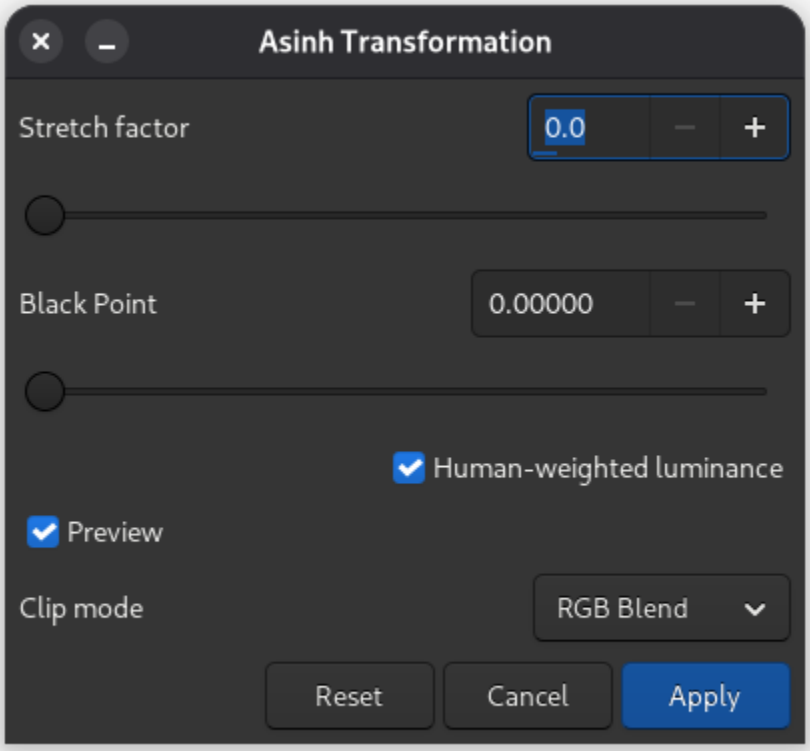


Abb. 2: Dialogbox der Asinh-Transformation

Für Farbbilder lautet die Funktion:

$$\text{pixel} = \frac{(\text{original} - \text{blackpoint}) \times \text{asinh}(\text{rgb_original} \times \text{stretch})}{\text{rgb_original} \times \text{asinh}(\text{stretch})}$$

wobei rgb_original unter Verwendung der Pixelwerte der drei Kanäle berechnet wird.

A clipping mode can also be set.

Theorie

Da rgb_original ein Durchschnitt der drei Kanäle ist, sind ein oder zwei Kanalwerte größer als rgb_original und können daher abgeschnitten werden. Dies kann zu Farbartefakten führen, wenn helle, farbintensive Bereiche gestreckt werden. Um dieses Problem zu vermeiden, wurde von den Autoren des ursprünglichen GHSastro-Tools der RGB-Blend-Clipping-Algorithmus entwickelt: Die gleichen Algorithmen sind in der Siril-Implementierung verfügbar, und dies ist der Standard-Clipping-Modus für Streckungen, die eine Clipping-Behandlung erfordern. Die Werte (r, g, b) werden zunächst auf der Grundlage des Luminanzwertes rgb_original gestreckt, um (r', g', b') zu erhalten. Dann werden die ursprünglichen Werte (r, g, b) unabhängig voneinander gestreckt, um $(r'\text{fl}, g`', b'')$ zu erhalten. Schließlich wird der größte Wert von k so identifiziert, dass

$k \times r' + (1 - k) \times r''1;$

$k \times g' + (1 - k) \times g''1;$

und

$k \times b' + (1 - k) \times b''1$

Dann werden die transformierten Werte wie folgt berechnet

$(k \times r' + (1 - k) \times r'', k \times g' + (1 - k) \times g'', k \times b' + (1 - k) \times b'')$

Dieser Algorithmus zum Beschneiden von RGB-Überblendungen wird auch für die weiter unten beschriebenen Generalisierten Hyperbolischen Streckungstransformationen verwendet.

Es stehen weitere Optionen für den Clipping-Algorithmus zur Verfügung:

- *Clip* - this clipping mode just allows any colour components that clip to clip, but restricts them to values in the range 0.0 to 1.0. This may suffer from coloured artefacts such as fringes around nearly-saturated stars, but it is extremely quick to calculate.
- *Rescale* - this clipping mode checks the R, G and B components of each pixel and if any are > 1.0 it rescales the pixel so that no components are clipped. This method is prone to artefacts and is mainly included for feature equivalence with the GHSastro plugin. It is quick to compute.
- *Global Rescale* - this clipping mode behaves similarly to *Rescale* except that the scaling is computed globally instead of per-pixel. This avoids the kind of artefacts that Rescale can produce, but has a bigger impact on overall image brightness. This is faster than *RGB blending* to compute but slower than *Clip* or *Rescale*.

Wenn die Option „Human-weighted Luminance“ (Menschlich gewichtete Leuchtdichte) nicht aktiviert ist, entspricht rgb_original dem Mittelwert der drei Pixelwerte. Wenn diese Option aktiviert ist, ändert sich die Gewichtung auf 0,2126 für den Rotwert, 0,7152 für den Grünwert und 0,0722 für den Blauwert, wodurch Ergebnisse erzielt werden, die der menschlichen Farbwahrnehmung näher kommen.

Siril Kommandozeile

```
asinh [-human] stretch { [offset] [-clipmode=] }
```

Streckt das Bild, um schwache Objekte mithilfe einer hyperbolischen Arcussinus-Transformation anzuzeigen. Das obligatorische Argument **stretch**, normalerweise zwischen 1 und 1000, gibt die Stärke der Streckung an. Der Schwarzpunkt kann verschoben werden, indem ein optionales Argument **offset** im normalisierten Pixelwert von [0, 1] angegeben wird. Schließlich ermöglicht die Option **-human** die Verwendung der Lichtempfindlichkeit des menschlichen Auges zur Berechnung der Leuchtdichte, die zur Berechnung des Streckungswerts für jedes Pixel verwendet wird, anstelle des einfachen Mittelwerts der Kanalpixelwerte. Diese Streckungsmethode bewahrt die Helligkeit des L*a*b*-Farbraums. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend.

10.1.2 Transformation Mitteltonübertragungsfunktion (MTF)

MTF ist eines der leistungsfähigsten Werkzeuge zum Strecken des Bildes. Sie kann leicht automatisiert werden, und deshalb wird sie auch in der automatisch gestreckten Ansicht verwendet.

Das Werkzeug wird in Form eines Histogramms mit 3 Schiebereglern (in Form von Dreiecken unter dem Histogramm) dargestellt, die wir bewegen müssen, um das Bild zu verändern. Das Dreieck auf der linken Seite steht für den Schattenpunkt, das Dreieck auf der rechten Seite für die Lichter und das Dreieck in der Mitte für die Mitteltonbalance. Die Werte dieser Schieberegler werden unterhalb des Histogramms auf der linken Seite angezeigt und können direkt von Hand verändert werden. Auf der anderen Seite steht der Prozentsatz der Pixel, die durch die Transformation abgeschnitten werden: Es ist wichtig, nicht zu viele Pixel abzuschneiden. Wenn nur der Parameter für die Mitteltöne geändert wird, kann kein Pixel abgeschnitten werden.

Theorie

Die neuen Pixelwerte werden dann mit dieser Funktion berechnet:

$$\mathrm{MTF}(x_p) = \frac{(m-1)x_p}{(2m-1)x_p - m}. \tag{10.1}$$

- Für $x_p = 0$, $\mathrm{MTF} = 0$,
- Für $x_p = m$, $\mathrm{MTF} = 0.5$,
- Für $x_p = 1$, $\mathrm{MTF} = 1$,

wobei x_p der Pixelwert ist wie folgt

$$x_p = \frac{\text{original} - \text{shadows}}{\text{highlights} - \text{shadows}}. \tag{10.2}$$

Bemerkung

Es wird im Allgemeinen nicht empfohlen, den Wert der Lichter zu ändern, da diese sonst gesättigt werden und Informationen verloren gehen.

Die Symbolleiste enthält viele Schaltflächen, die sich auf die Darstellung des Histogramms auswirken. Sie können wählen, ob das Eingabehistogramm, das Ausgabehistogramm, die Übertragungskurve und das Raster angezeigt werden sollen. Mit der Schaltfläche können Sie dieselbe Transformation wie beim Autostretch-Algorithmus anwenden. Es ist selten ratsam, diese Schaltfläche unverändert zu verwenden. In der Regel sind Anpassungen erforderlich, um Informationsverluste zu vermeiden.

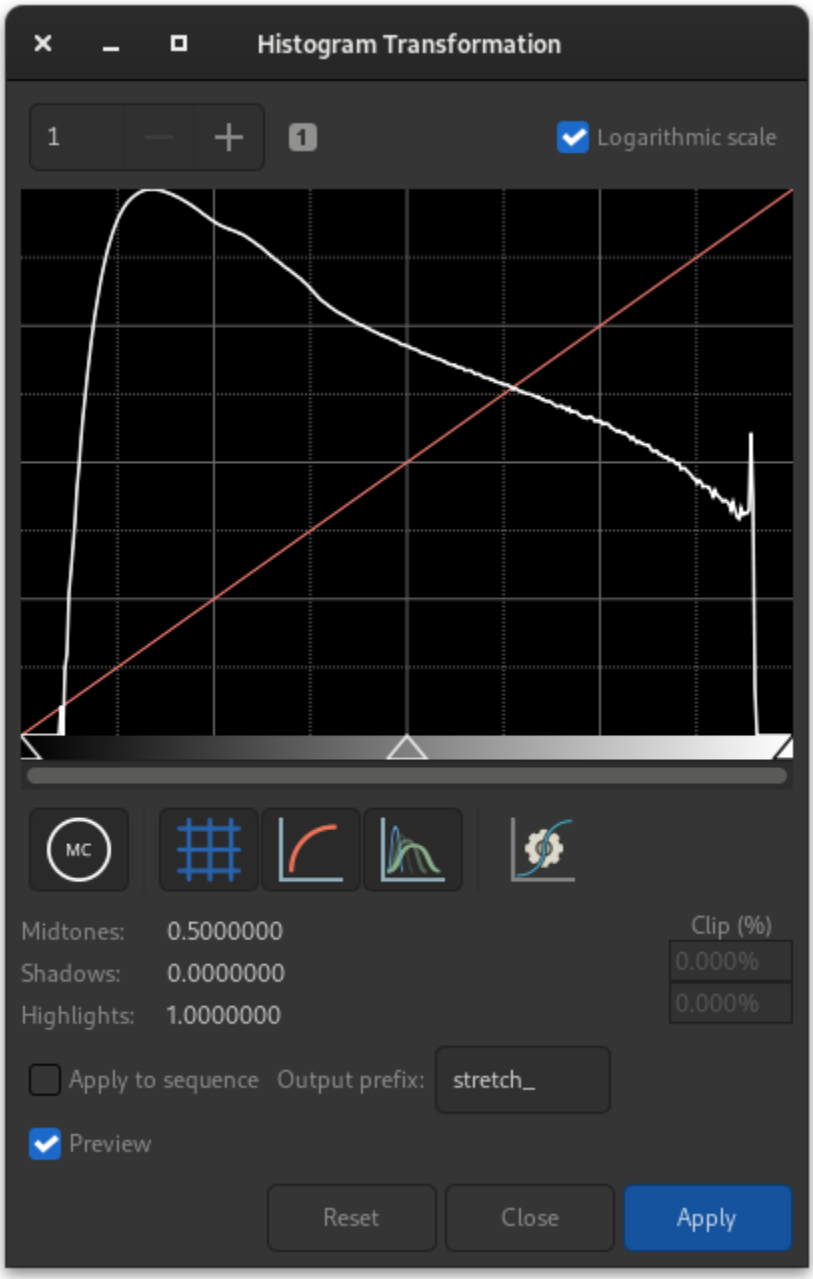


Abb. 3: Dialogbox der Histogramm-Transformation

Tipp

When the button is pressed, the sliders and lo, mid and hi entries will become temporarily inactive. You have to apply the autostretch with the *Apply* button and the controls will reactivate. You can then apply adjustments as a second follow-up stretch. This behaviour avoids problems where the monitor ICC profile is different to the image ICC profile.

Am oberen Rand des Histogramms kann auch die logarithmische Ansicht ausgewählt werden, wie in der Abbildung gezeigt. Diese Einstellung kann wie unter *hier* beschrieben als Standard festgelegt werden. Schließlich steht eine Vergrößerung in X-Richtung zur Verfügung. Dies ist sehr nützlich, wenn sich das gesamte Signal auf der linken Seite des Histogramms konzentriert.

Tipp

If a ROI is set, the MTF histogram preview will not update to show the impact of the stretch on the ROI. This is because that behaviour could be misleading: if the ROI is not typical of the image overall, adjusting the ROI histogram to a suitable level would result in a badly adjusted histogram for the overall image and potentially a burned-out or excessively dark look to the result. When in ROI mode the stretch parameters should be adjusted by eye. If it is desired to check the histogram for the stretch as applied to the image as a whole, the ROI should be cleared.

Siril Kommandozeile

```
mtf low mid high [channels]
```

Wendet die Mittelton-Übertragungsfunktion auf das aktuell geladene Bild an.

Es werden drei Parameter benötigt: **low**, **midtones** und **high**, wobei der Parameter für die Mitteltonbalance eine nichtlineare Histogrammstreckung im Bereich [0,1] definiert. Für eine automatische Ermittlung der Parameter siehe AUTOSTRETCH.
Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: dies können R, G, B, RG, RB oder GB sein. Die Standardeinstellung ist alle Kanäle

Verweise: *autostretch*

Bemerkung

`mtf` ist auch eine Funktion, die in dem Werkzeug *PixelMath* verwendet werden kann.

Siril Kommandozeile

```
autostretch [-linked] [shadowsclip [targetbg]]
```

Streckt das aktuell geladene Bild mit unterschiedlichen Parametern für jeden Kanal (nicht verknüpft), sofern nicht **-linked** übergeben wird. Argumente sind optional, **shadowclip** ist der Schatten-Beschneidungspunkt, gemessen in Sigma-Einheiten von der Spitze des Haupthistogramms (Der Defaultwert ist -2,8), **targetbg** ist der Ziel-Hintergrundwert, der dem Bild seine endgültige Helligkeit gibt. Bereich [0, 1], Defaultwert ist 0,25. Die Defaultwerde werden in der Auto-Stretch-Funktion der grafischen Benutzeroberfläche benutzt.

Verwenden Sie nach der Farbkalibrierung nicht die nicht verknüpfte Version, da sie den Weißabgleich verändert

Anwendung der Transformation auf die Sequenz

Diese Transformation kann leicht auf eine Sequenz angewendet werden. Sie müssen nur die Transformation für das geladene Bild (mit einer bereits geladenen Sequenz) definieren, dann die Schaltfläche *Auf Sequenz anwenden* aktivieren und das Ausgabepräfix der neuen Sequenz (standardmäßig `mtf_`) definieren, oder den folgenden Befehl verwenden:

Siril Kommandozeile

```
seqmtf sequencename low mid high [channels] [-prefix=]
```

Derselbe Befehl wie MTF, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "mtf_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweis: *mtf*

10.1.3 Verallgemeinerte hyperbolische Streckungstransformationen (Generalised Hyperbolic Stretch/GHS)

Dies ist das leistungsfähigste und modernste Werkzeug von Siril, aber auch das am schwierigsten zu erlernende. Eine sehr ausführliche Anleitung für dieses Tool in Siril wurde von den Autoren dieses Algorithmus geschrieben: https://siril.org/tutorials/ghs. Wir werden hier nur die grundlegende Funktionsweise dieses Tools zusammenfassen.

Einfach ausgedrückt, kann das GHS den Kontrast eines Helligkeitsbereichs in einem Bild verbessern. Wenn man zum Beispiel die Details im mittleren bis hohen Helligkeitsbereich eines Nebels (der im Allgemeinen in einem Astrobild sehr schwach ist) besser sehen möchte, kann man nur diesen Bereich für die Streckung auswählen. Es ist sehr gut geeignet, den Kontrast von Hauptobjekten zu verbessern, ohne dass die Sterne zu groß werden. Das Werkzeug basiert auf einer iterativen Anwendung, d. h. es werden nacheinander alle Helligkeitsbereiche des Bildes durch kleine Eingriffe gestreckt.

Um dies zu erreichen, stützt sich das Werkzeug in hohem Maße auf die Anzeige und Interaktion von Histogrammen für jeden Farbkanal. Die Transformationsfunktion, die die Form einer Hyperbel oder eines "S" hat, kann durch Verschieben ihres Zentrums (Parameter **SP** - Symmetriepunkt), durch Abflachen eines ihrer Enden (mit **Schatten-Schutzpunkt (SP)** und **Lichter-Schutzpunkt (HP)**) und natürlich durch ihre Verdrehung (Faktoren **D** und **b** für die lokale Dehnung) verändert werden. Die Manipulation dieser Parameter an einem kleinen Bild (für die Geschwindigkeit) mit einem **SP**-Wert von 0,5 wird Ihnen helfen, ihre Wirkung zu verstehen.

Bei jeder Iteration gibt es zwei Hauptoperationen: die Auswahl des zu ändernden Helligkeitsbereichs und die eigentliche Änderung des Bereichs. Die Auswahl des Bereichs ist recht einfach, es geht darum, einen repräsentativen Wert (**SP**) zu finden und die Breite des Bereichs (**b**) zu definieren. Die Einstellung von **SP** kann auf drei Arten erfolgen:

- Auswahl eines Bereichs mit ähnlicher Helligkeit im Bild und Klicken auf die Schaltfläche "Pipette"

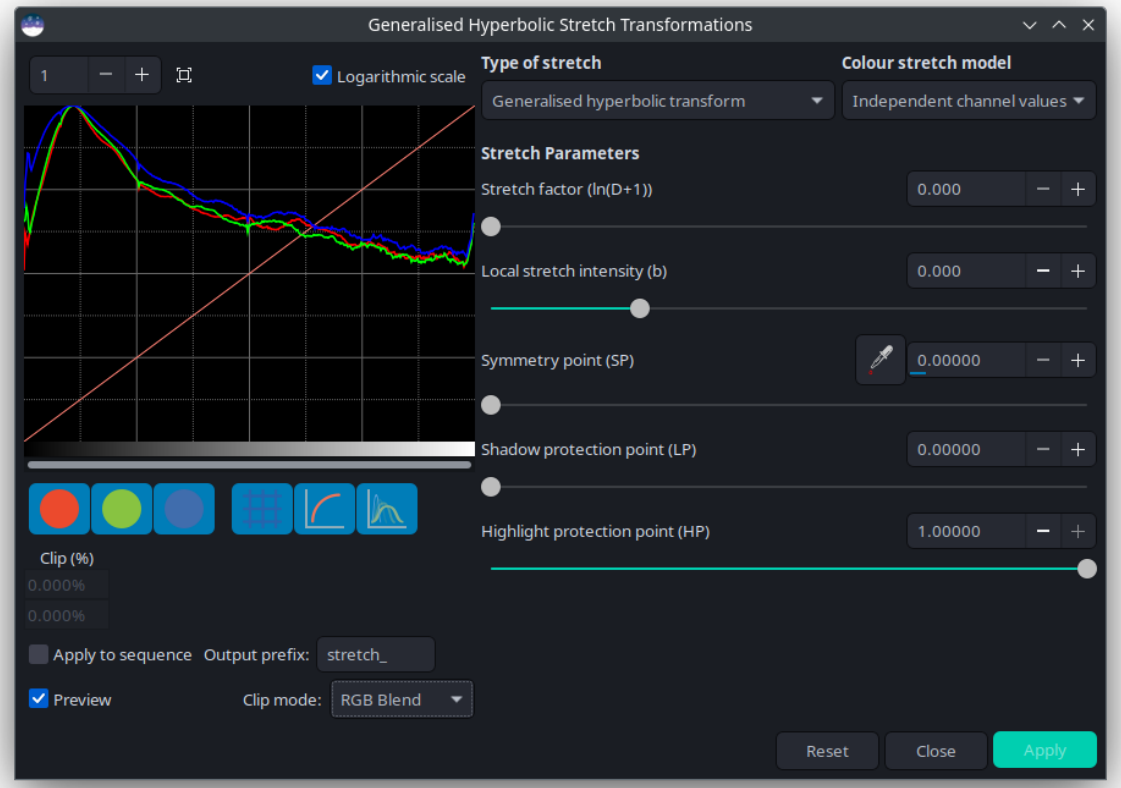


Abb. 4: Dialogbox für die Allgemeine Hyperbolische Streckung

- Anklicken des Histogramms mit der linken Maustaste (es ist möglich, das Histogramm mit Hilfe der Schaltfläche + oben links zu vergrößern)
- mit dem Cursor oder den zugehörigen Plus- und Minus-Tasten oder der direkten Eingabe eines Wertes.

Die Breite des Bereichs hängt von der lokalen Dehnung ab. Ein hoher Wert von **b** ergibt einen kleinen Bereich und erhöht den Kontrast in einem kleinen Helligkeitsbereich des Bildes.

Die Änderung des Histogramms, nachdem der Ort der Änderung festgelegt wurde, ist ein komplexerer Vorgang. Ein von den Autoren des Algorithmus vorgegebenes Ziel ist es, die logarithmische Ansicht des Histogramms (aktiviert durch Ankreuzen des Kästchens) so nah wie möglich an eine abfallende Linie heranzuführen. Zu diesem Zweck müssen Unebenheiten beseitigt und Täler aufgefüllt werden. Im Folgenden finden Sie eine kurze Übersicht über die zu verwendenden Werte, je nachdem, was erreicht werden soll:

- **Anfängliche Streckung von linear: SP** leicht links vom Hauptpeak einstellen, **b** Wert von 6 und höher moderieren, **D** leicht erhöhen, nur um das Hauptobjekt zu sehen. Dehnen Sie zu diesem Zeitpunkt nicht zu sehr, wie es ein Autostretch tun würde, sonst würden die Sterne zu groß werden (<https://siril.org/tutorials/ghs/#performing-initial-stretches-using-ghs>`_).
- **Verbesserung des Kontrasts eines Bereichs oder Auffüllen eines Tals:** Stellen Sie **SP** auf die Mitte des Tals im Histogramm ein, stellen Sie **b** so hoch ein, wie der Bereich oder das Tal schmal ist, verringern Sie **HP**, um die Sterne nicht aufzublähen, erhöhen Sie **D** langsam, bis die Verbesserung sichtbar wird.
- **Verringerung des Kontrasts eines Bereichs oder Abflachung eines Spitzenwerts:** Die Verringerung eines Spitzenwerts ist nicht einfach zu bewerkstelligen, erfolgt aber als Nebeneffekt der Auffüllung von Tälern. So wird beispielsweise durch das Erzeugen einer Spitze oder das Auffüllen eines Tals der Bereich links von **SP** verringert. Eine andere Möglichkeit ist die Verwendung der inversen Transformation aus dem Kombinationsfeld *Typ der Streckung*, ein hoher **LP**-Wert und **HP** bei 1.
- **Kurve nach links verschieben, wodurch das Bild dunkler wird:** Wenn wir das gesamte Histogramm strecken, verschiebt sich die Spitze oft nach rechts, wodurch der Hintergrund zu hell wird. Es gibt eine einfache Möglichkeit, alles nach links zu verschieben: Wählen Sie im Kombinationsfeld *Typ der Streckung* den letzten Eintrag, **Lineare Streckung (Schwarzpunkt-Verschiebung)**. Jetzt muss nur noch ein Cursor bewegt werden, der bestimmt, wie weit er verschoben wird - oder Sie klicken im Histogramm auf den gewünschten Schwarzpunkt links neben dem Berg.

Tipp

If a ROI is set, the GHT histogram preview will not update to show the impact of the stretch on the ROI. This is because that behaviour could be misleading: if the ROI is not typical of the image overall, adjusting the ROI histogram to a suitable level would result in a badly adjusted histogram for the overall image and potentially a burned-out or excessively dark look to the result. When in ROI mode the stretch parameters should be adjusted by eye. If it is desired to check the histogram for the stretch as applied to the image as a whole, the ROI should be cleared.

Einige Operationen sind auch bei **Farbbildern** üblich, da wir oft eine ähnliche Kurvenform für die drei Kanäle haben wollen. Wir können jeden Kanal unabhängig bearbeiten, indem wir sie mit den drei farbigen Kreisen unter der Histogrammansicht abwählen:

- **Verschieben der Spitze nach rechts:** eine einfache Streckung mit einem **SP**-Wert links von der Spitze wird dies im Allgemeinen bewirken, daher sollte dies als Teil einer Streckung erfolgen.
- **Einen Berg strecken:** um einen Kanal etwas mehr zu strecken und ihm mehr Bedeutung im Endergebnis zu geben, ohne die Position des Peaks zu sehr zu verändern, stellen Sie **SP** in der Nähe der Spitze oder leicht rechts davon ein, stellen Sie **b** ein, je nachdem, wie der Beitrag im gesamten Kanal gewünscht wird, zwischen einem negativen Wert, wenn die Auswirkung bis zu den Sternen zu spüren sein soll (um ihre Farbe zu ändern) und einem hohen Wert, wenn dies nur für einen Nebel gilt, erhöhen Sie **D**, um die Zielbreite der Spitze zu erhalten, und versetzen Sie dann den Berg nach links, indem Sie **HP** erhöhen.

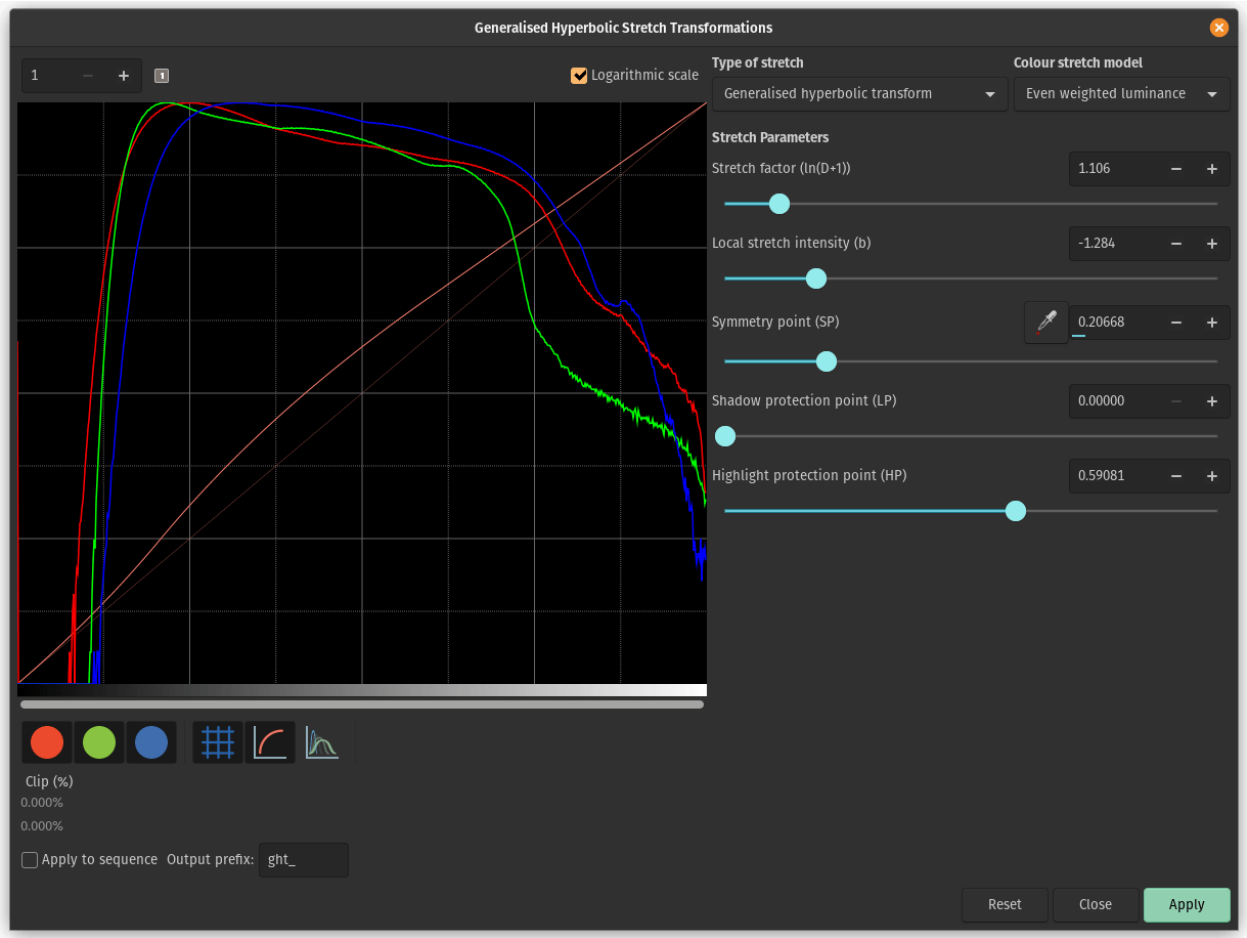


Abb. 5: Die generalisierte hyperbolische Streckung für ein Farbbild

- **Alle Kanäle zusammen verschieben:** Es gibt eine alternative Streckung der Luminanzzuordnung, siehe das Kombinationsfeld *Farbstreckungsmodell* oben rechts im GHS-Fenster. Wenn Sie einen der beiden Luminanz-Streckungswerte verwenden, wird die Luminanz gestreckt und die Farben darauf neu angewendet, anstatt die drei Kanäle direkt zu strecken. Mit den Luminanzmodi können die Farben im Bild besser erhalten werden. Diese Modi verwenden denselben RGB-Überblendungsmodus, der oben beschrieben wurde, um Artefakte beim Beschneiden der Farbkanäle zu vermeiden.
- **Bildsättigung neu abbilden:** Die GHS-Transformationen können auf den Sättigungskanal des Bildes angewendet werden, indem die Option Sättigung aus dem Kombinationsfeld *Farbdehnungsmodell* ausgewählt wird. Wenn dieser Modus ausgewählt ist, werden die Histogramme der Sättigung vor und nach der Streckung gelb angezeigt. Alle GHS-Optionen sind verfügbar und dieser Modus ermöglicht eine sehr gezielte Anpassung des Sättigungskanals. Eine einfache Methode, um die Sättigung in relativ ungesättigten Regionen zu erhöhen und gleichzeitig eine Übersättigung zu vermeiden, ist die Verwendung einer **Inversen verallgemeinerten hyperbolischen Transformation**, bei der **SP** auf etwa 0,5 eingestellt ist und **HP** so weit heruntergefahren wird, dass das obere Ende des Sättigungshistogramms abgeflacht wird.

In order to control partially-clipped highlights, the GHS tool makes available the same range of clipping modes as the asinh stretch. Details can be found *here*.

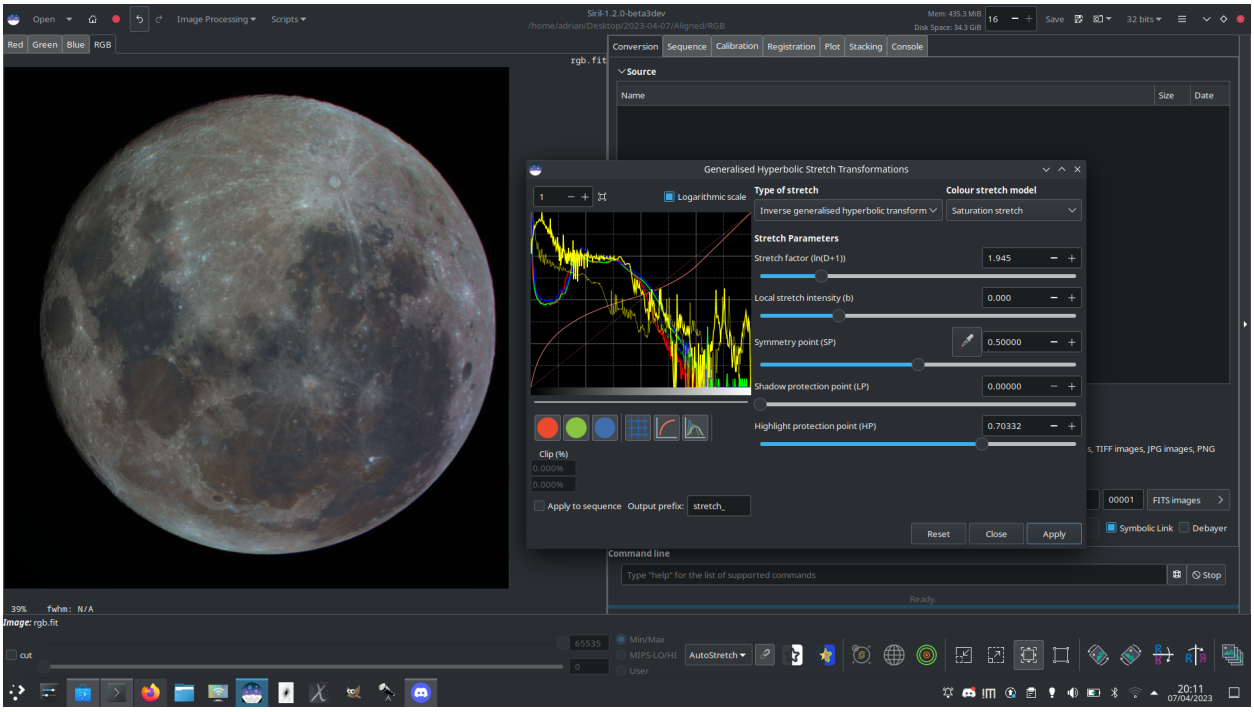


Abb. 6: Das obige Bild zeigt, wie die Anwendung des GHS-Werkzeugs auf den Sättigungskanal eine einfache Möglichkeit bietet, die Sättigung in einem Bild mit geringer Sättigung stark zu verstärken und gleichzeitig die Kontrolle über das obere Ende des Sättigungshistogramms zu behalten, das hier verwendet wurde, um ein "Mineralische Mond"-Bild zu erstellen, das die unterschiedliche mineralische Zusammensetzung der verschiedenen Regionen der Mondoberfläche hervorhebt.

Siril Kommandozeile

```
ght -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -
↪sat] [channels]
```

Verallgemeinerte hyperbolische Streckung (Generalized Hyperbolic Stretch) auf der Grundlage der Arbeit des Teams von ghsastro.co.uk.

Das Argument **-D=** bestimmt die Stärke der Streckung, die zwischen 0 und 10 liegt. Dies ist das einzige obligatorische Argument. Mit den folgenden optionalen Argumenten kann die Streckung weiter angepasst werden:

B bestimmt die Intensität der Streckung in der Nähe des Symmetriepunktes, zwischen -5 und 15;

LP definiert einen schattenerhaltenden Bereich zwischen 0 und SP, in dem die Streckung linear erfolgt und die Schattendetails erhalten bleiben;

SP definiert den Symmetriepunkt der Streckung, der zwischen 0 und 1 liegt und an dem die Streckung am stärksten ist;

HP definiert einen Bereich zwischen HP und 1, in dem die Streckung linear erfolgt, wodurch die Details der Glanzlichter erhalten bleiben und eine Aufblähung der Sterne verhindert wird.

Wenn B, LP und SP weggelassen werden, ist der Standardwert 0,0 und bei HP ist der Standardwert 1,0.

Ein optionales Argument (entweder **-human**, **-even** oder **-independent**), kann übergeben werden, um entweder menschlich-gewichtete oder gleichmäßig gewichtete Luminanz oder unabhängige Farbkanäle für Farbstreckungen auszuwählen. Das Argument wird bei Monobildern ignoriert. lternativ gibt das Argument **-sat** an, dass die Streckung anhand der Bildsättigung erfolgt - das Bild muss farbig sein und alle Kanäle müssen ausgewählt sein, damit dies funktioniert.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: Dies können R, G, B, RG, RB oder GB sein. Der Standardwert sind alle Kanäle. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte wie **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend.

Siril Kommandozeile

```
invght -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -
↪sat] [channels]
```

Kehrt eine generalisierte hyperbolische Streckung um. Es bietet die umgekehrte Transformation von GHT. Wenn es mit denselben Parametern versehen wird, macht es einen GHT-Befehl rückgängig und kehrt möglicherweise zu einem linearen Bild zurück. Es kann für Negativbilder auch auf die identische Weise wie GHT funktionieren

Verweise: *ght*

Siril Kommandozeile

```
modasinh -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -sat]␣
↪[channels]
```

Modifizierte ArcSinH-Streckung auf der Grundlage der Arbeit des Teams von ghsastro.co.uk.

Das Argument **-D=** bestimmt die Stärke der Streckung, die zwischen 0 und 10 liegt. Dies ist das einzige obligatorische Argument. Mit den folgenden optionalen Argumenten kann die Streckung weiter angepasst werden:

LP definiert einen schattenerhaltenden Bereich zwischen 0 und SP, in dem die Streckung linear erfolgt und die Schattendetails erhalten bleiben;

SP definiert den Symmetriepunkt der Streckung, der zwischen 0 und 1 liegt und an dem die Streckung am stärksten ist;

HP definiert einen Bereich zwischen HP und 1, in dem die Streckung linear erfolgt, wodurch die Details der Glanzlichter erhalten bleiben und eine Aufblähung der Sterne verhindert wird.

Ohne diese Angabe haben LP und SP den Standardwert 0,0 und HP den Standardwert 1,0.

Ein optionales Argument (entweder **-human**, **-even** oder **-independent**), kann übergeben werden, um entweder menschlich-gewichtete oder gleichmäßig gewichtete Luminanz oder unabhängige Farbkanäle für Farbstreckungen auszuwählen. Das Argument wird bei Monobildern ignoriert. lternativ gibt das Argument **-sat** an, dass die Streckung anhand der Bildsättigung erfolgt - das Bild muss farbig sein und alle Kanäle müssen ausgewählt sein, damit dies funktioniert.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: Dies können R, G, B, RG, RB oder GB sein. Der Standardwert sind alle Kanäle. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte wie **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend.

Siril Kommandozeile

```
invmodasinh -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -
↪sat] [channels]
```

Kehrt eine modifizierte Arcsinh-Streckung um. Es bietet die umgekehrte Transformation von MODASINH. Wenn es mit denselben Parametern versehen wird, macht es einen MODASINH-Befehl rückgängig und kehrt möglicherweise zu einem linearen Bild zurück. Es kann auch bei Negativbildern auf die gleiche Weise wie MODASINH funktionieren

Verweise: *modasinh*

Siril Kommandozeile

```
linstretch -BP= [-sat] [-clipmode=] [channels] [-clipmode=]
```

Streckt das Bild linear auf einen neuen Schwarzpunkt BP.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: dies können R, G, B, RG, RB oder GB sein. Die Standardeinstellung ist alle Kanäle

Optional kann der Parameter **-sat** verwendet werden, um die lineare Streckung auf den Bildsättigungskanal anzuwenden. Dieses Argument funktioniert nur, wenn alle Kanäle ausgewählt sind. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend

Anwendung der Transformation auf die Sequenz

Diese Transformation kann leicht auf eine Sequenz angewendet werden. Sie müssen nur die Transformation auf das geladene Bild (mit einer bereits geladenen Sequenz) definieren, dann die Schaltfläche *Auf Sequenz anwenden* aktivieren und das Ausgabepräfix der neuen Sequenz definieren (standardmäßig `stretch_`). Alle Befehle haben auch eine Form der Sequenzverarbeitung. Jeder Sequenzstreckungsbefehl beginnt mit *seq* und das erste Argument muss der Sequenzname sein, aber ansonsten sind sie gleich.

Siril Kommandozeile

```
seqght sequence -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie GHT, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *ght*

Siril Kommandozeile

```
seqinvght sequence -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie INVGHT, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *invght*

Siril Kommandozeile

```
seqmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie MODASINH, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *modasinh*

Siril Kommandozeile

```
seqinvmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie INVMODASINH, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *invmodasinh*

Siril Kommandozeile

```
seqlinstretch sequence -BP= [channels] [-sat] [-prefix=]
```

Gleicher Befehl wie LINSTRETCH, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *linstretch*

10.1.4 Kurventransformation

Curves Transformation is a highly versatile tool used to adjust the contrast and brightness of an image by modifying the pixel values according to a custom-defined curve. This allows for precise control over the image's stretch.

The curve is defined by a series of points, each of which can be moved to adjust the curve. The curve is interpolated between these points, and the pixel values are transformed based on it. This allows for a wide range of transformations to be applied to the image, from simple linear stretches to complex non-linear adjustments.

There are two interpolation algorithms available in the Curves Transformation dialog: linear and cubic spline.

Theorie

- Linear interpolation is a simple interpolation that connects the points with straight lines. For each pair of points, the slope of the line connecting the points is calculated:

$$\text{slope} = \frac{y_2 - y_1}{x_2 - x_1}. \tag{10.3}$$

  The pixel values are then transformed by evaluating the line at the original pixel value:

$$\text{pixel} = \text{slope} \times (\text{original} - x_1) + y_1. \tag{10.4}$$

- Cubic spline curves are more complex curves that are defined by a series of control points. For pixel value x between two control points x_i and x_{i+1}, the curve is defined by the following equation:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{10.5}$$

For $x_i < x < x_{i+1}$, the coefficients a_i, b_i, c_i, and d_i are calculated by solving a system of equations derived from the conditions of continuity and smoothness at each internal point. These conditions are:

- The spline must be continuous at each internal point,
- The first derivative of the spline must be continuous at each internal point,
- The second derivative of the spline must be continuous at each internal point
- Since Curves Transformation uses natural cubic splines, the second derivative at both endpoints is 0.

Abb. 7: Dialogbox der Kurven-Transformation

Warnung

Curves Transformation is only available as a GUI tool and cannot be used from the command line.

Curves Transformationcan display the histogram of the image in two modes: linear and logarithmic. You can swap between these modes by clicking the *Logarithmic scale* checkbox at the top of the histogram. Logarithmic scale is useful for images with a wide dynamic range, as it allows you to see the histogram in the shadows and highlights more clearly.

Tipp

If a ROI is set, the histogram preview will not update to show the impact of the stretch on the ROI. This is because that behaviour could be misleading: if the ROI is not typical of the image overall, adjusting the ROI histogram to a suitable level would result in a badly adjusted histogram for the overall image and potentially a burned-out or excessively dark look to the result. When in ROI mode the stretch parameters should be adjusted by eye. If it is desired to check the histogram for the stretch as applied to the image as a whole, the ROI should be cleared.

Some common uses of the Curves Transformation tool include:

- **"S" curve:** This curve is used to increase the contrast of an image. By increasing the slope of the curve in the middle of the histogram, the contrast is increased.

Abb. 8: An example of an "S" curve

- **BP & WP adjustments:** The black point (BP) and white point (WP) of an image can be adjusted by moving the first and last control points of the curve. This allows for the shadows and highlights of the image to be adjusted independently.

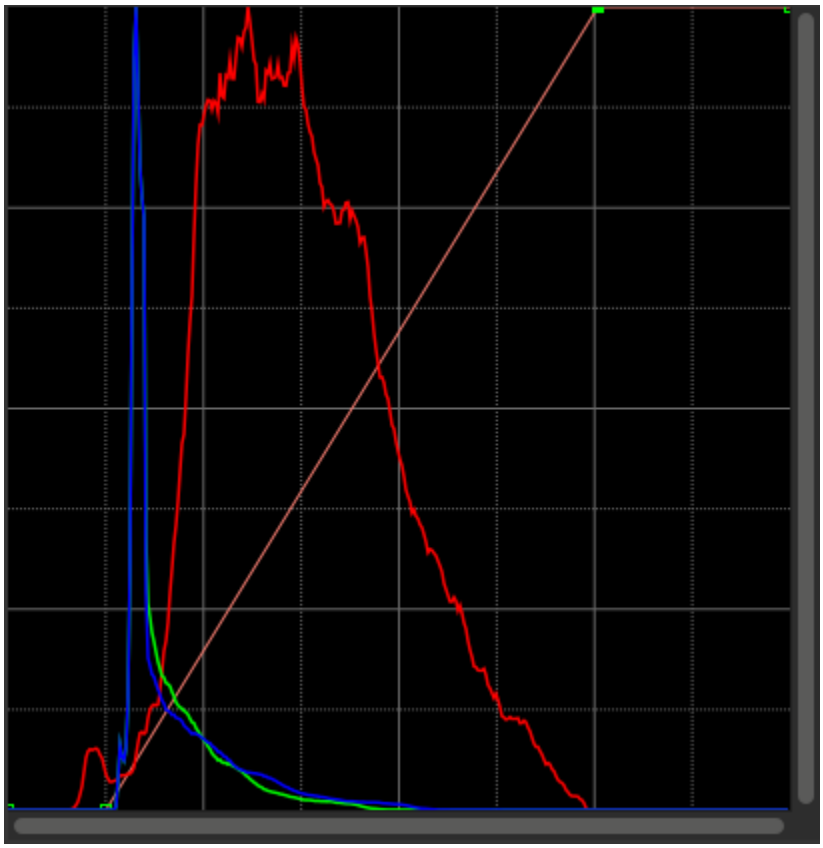


Abb. 9: An example of adjusting the black point and white point of an image

- **Targeted adjustments:** By adding control points at specific locations in the histogram, targeted adjustments can be made to the image. For example, the shadows can be darkened without affecting other parts of the image.

Abb. 10: An example of a targeted adjustment to an image

> **Tipp**
>
> The Curves Transformation tool is best used on an image that has already been stretched to some extent. This allows for more precise control over the image's contrast and brightness.

Diese Transformation kann leicht auf eine Sequenz angewendet werden. Sie müssen lediglich die Transformation für das aktuell geladene Bild definieren (mit einer bereits geladenen Sequenz), dann die Schaltfläche *Apply to sequence* aktivieren und das Ausgabepräfix der neuen Sequenz angeben (standardmäßig `curve_`).

10.2 Farbkalibrierung

Siril bietet drei Möglichkeiten, die Farben Ihres Bildes abzurufen:

- *Manualle Farbkalibrierung*
- *Photometrische Farbkalibrierung*, auch PCC genannt.
- ref:*Spektralphotometrische Farbkalibrierung <processing/color-calibration/spcc:Spectrophotometric Color Calibration>*, auch SPCC genannt. Dies ist die genaueste und empfohlene Methode

"Abrufen" bedeutet hier, dass die RGB-Kanäle neu abgeglichen werden, um den wahren Farben des aufgenommenen Objekts so nahe wie möglich zu kommen.

> **Tipp**
>
> PCC und SPCC sind zwei Methoden, die ein *astrometrisch gelöstes Bild* erfordern. Ohne diese Voraussetzung ist es nicht möglich, auf diese Werkzeuge zuzugreifen.

10.2.1 Farbkalibrierung (manuell)

> **Warnung**
>
> Die Farbkalibrierung **muss** an einem linearen Bild durchgeführt werden, dessen Histogramm noch nicht gestreckt wurde. Andernfalls sind die erhaltenen Farben nicht garantiert korrekt.

Bei der manuellen Methode wird das folgende Fenster verwendet:

Der erste Schritt befasst sich mit dem **Hintergrund** Ihres Bildes. Das Ziel ist es, die RGB-Ebenen anzugleichen, damit der Hintergrund als neutrale graue Farbe erscheint.

Nachdem Sie eine Auswahl in Ihrem Bild festgelegt haben (in einem nicht zu kleinen oder zu kontrastreichen Bereich), wird der Bereich berücksichtigt, indem Sie auf die Schaltfläche *Verwende aktuelle Auswahl* im oberen Bereich klicken. Die Koordinaten des Rechtecks werden angezeigt. Dann berechnet *Hintergrund Neutralisation* den Median jedes Kanals und gleicht ihn aus.

Der zweite Schritt befasst sich mit den **hellen Objekten** des Bildes. Auch hier können Sie das Histogramm auf zwei Arten verändern:

- Manuell, mit **Weiß-Referenz** und den 3 Koeffizienten R, G und B, je nach eigenem Geschmack.
- Automatisch, durch Auswahl eines rechteckigen Bereichs mit kontrastreichen Objekten (wie zuvor) durch Klick auf die untere Schaltfläche *Verwende aktuelle Auswahl*

Mit zwei Schiebereglern können Sie die Ausschlussgrenze für zu dunkle und zu helle Pixel in der Auswahl ändern.

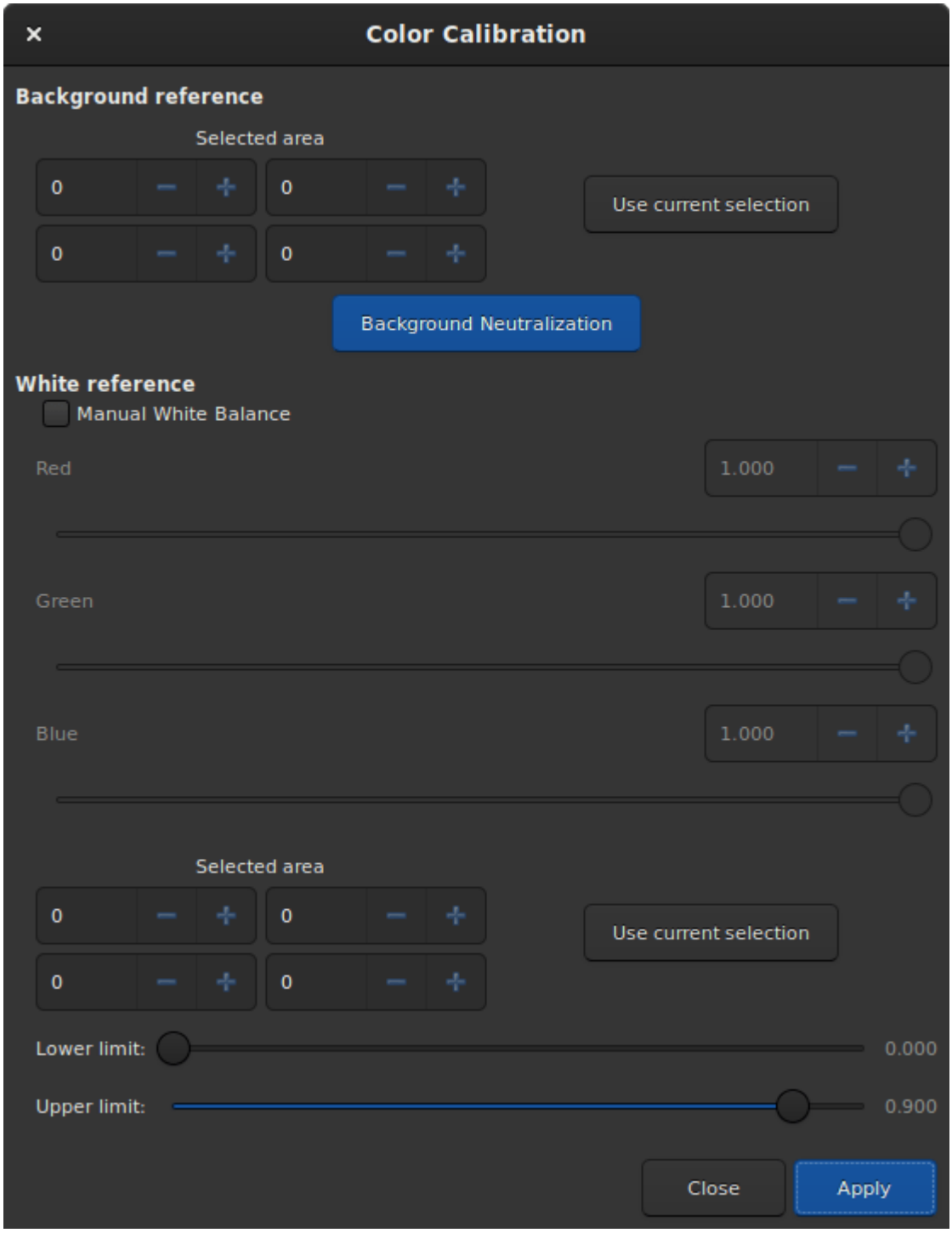


Abb. 11: Dialogfenster Farbkalibrierung.

Da es sich hierbei um einen Prozess mit Versuch und Irrtum handelt, können Sie das Ergebnis mit der Schaltfläche *Undo* (oben links) rückgängig machen und es dann mit anderen Auswahlbereichen oder Koeffizienten versuchen, bis Sie zufrieden sind.

10.2.2 Photometrische Farb-Kalibrierung

Warnung

Die Kalibrierung der Farben durch Photometrie **muss** unbedingt an einem linearen Bild durchgeführt werden, dessen Histogramm noch nicht gestreckt wurde. Andernfalls wird die photometrische Messung falsche Ergebnisse liefern und es gibt keine Garantie für die Korrektheit der erhaltenen Farben.

Eine andere Möglichkeit, die Farben abzurufen, besteht darin, die Farbe der Sterne im Bild mit ihrer Farbe in Katalogen zu vergleichen, um automatisch und nicht subjektiv die natürlichste Farbe zu erhalten. Dies ist das PCC-Tool (Photometric Color Calibration) (`Strg` + `Shift` + `P`). Dies funktioniert nur für Bilder, die mit einem Satz roter, grüner und blauer Farbfilter oder mit Farbbildern eines RGB-Sensors aufgenommen wurden. Um Sterne im Bild mit denen des Katalogs zu identifizieren, ist eine astrometrische Lösung erforderlich. Bitte beachten Sie die *Dokumentation des Lösermoduls*.

Diese Methode ist weniger genau als die unten erläuterte spektralfotometrische Farbkalibrierung, kann jedoch mithilfe lokaler Kataloge durchgeführt werden und ist daher die beste Option, wenn keine Internetverbindung verfügbar ist.

Bemerkung

Diese Technik ist stark von der Art des verwendeten Filters abhängig. Die Verwendung verschiedener Arten von R-, G- und B-Filtern macht keinen großen Unterschied, aber die Verwendung eines Lichtverschmutzungsfilters oder das fehlen eines IR-Sperrfilters führt dazu, dass die Lösung erheblich abweicht und nicht die erwarteten Farben liefert.

Seit Version 1.4 laufen die beiden Tools unabhängig voneinander: Die photometrische Analyse und Farbkorrektur des Bildes ist nur dann möglich, wenn das Bild bereits astrometrisch gelöst wurde. Dies bedeutet auch, dass für PCC und Astrometrie unterschiedliche Kataloge verwendet werden können. Das Tool ist auch als `pcc`-Befehl verfügbar, sodass es in Bildnachbearbeitungsskripte eingebettet werden kann.

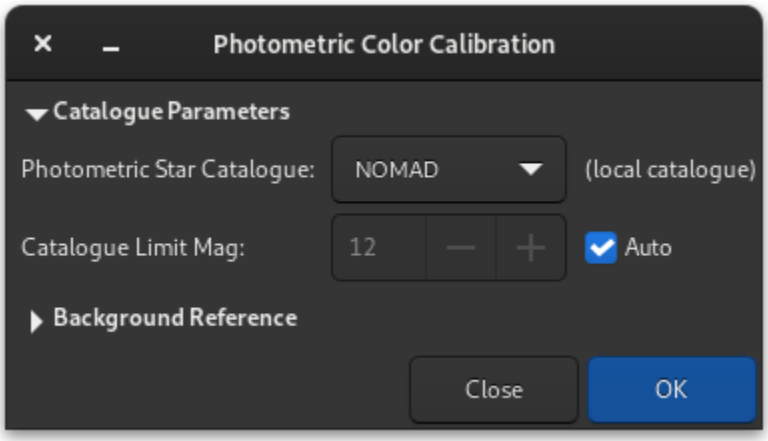


Abb. 12: Dialogfenster Photometrische Farbkalibrierung.

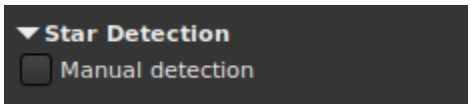
Wenn das Bild zuvor astrometrisch gelöst wurde, aktivieren Sie die Funktion „:ref:`Beschriftung <astrometry/annotations:annotations>“, um zu überprüfen, ob die Kataloge mit dem Bild übereinstimmen. Wenn die astrometrische Lösung nicht gut genug ist, wiederholen Sie bitte die die astrometrische Lösung.

- Im Abschnitt **Katalogeinstellungen** können Sie auswählen, welcher photometrische Katalog verwendet werden soll, NOMAD, APASS oder Gaia DR3, sowie die Grenzgröße.

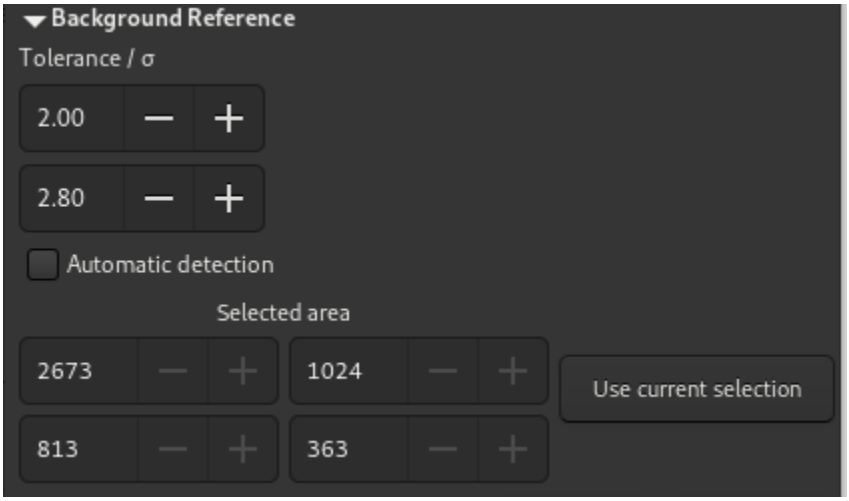
> **Tipp**
>
> Der NOMAD und GAIA-Katalog kann *lokal installiert* werden, während der APASS-Katalog eine Internetverbindung benötigt.
>
> The local Gaia catalog used for PCC is the astrometric extract, which is a much smaller download than the whole SPCC catalog. If this is available it will be shown as the default.

> **Bemerkung**
>
> Whereas the NOMAD and APASS catalogs use the catalogued B-V index and estimate the effective temperature from this value, Gaia provides an accurately modelled effective temperature therefore when this catalog is used the Teff field is used directly, giving more accurate results.

- Im Abschnitt **Sternenerkennung** können Sie manuell auswählen, welche Sterne für die photometrische Analyse verwendet werden sollen. Es ist besser, mindestens Hunderte von ihnen zu haben, so dass eine individuelle Auswahl nicht ideal wäre.



- Falls gewünscht, kann die **Hintergrundreferenz** manuell ausgewählt werden, wie in *Manuelle Farbkalibrierung* beschrieben. Dies kann bei Bildern großflächiger Nebel nützlich sein, bei denen nur sehr wenig Himmelshintergrund sichtbar ist.



Wenn genügend Sterne gefunden wurden und die astrometrische Lösung korrekt ist, druckt der PCC diese Art von Text auf der Registerkarte Konsole aus:

```
Applying aperture photometry to 433 stars.
70 stars excluded from the calculation
Distribution of errors: 1146 no error, 18 not in area, 48 inner radius too small, 4␣
↪pixel out of range
Found a solution for color calibration using 363 stars. Factors:
K0: 0.843       (deviation: 0.140)
K1: 1.000       (deviation: 0.050)
K2: 0.743       (deviation: 0.130)
The photometric color correction seems to have found an imprecise solution, consider␣
↪correcting the image gradient first
```

Wir sehen, dass 433 Sterne aus dem Katalog und dem Bild für die photometrische Analyse ausgewählt wurden, aber irgendwie haben wir nur 363 tatsächlich verwendet, 70 wurden ausgeschlossen. Die Zeile *Distribution of errors* erklärt, warum sie ausgeschlossen wurden: 18 wurden nicht an der erwarteten Position gefunden, 48 waren zu groß und 4 wahrscheinlich gesättigt. Es kommt sehr häufig vor, dass viele Sterne ausgeschlossen werden, weil sie die strengen Anforderungen für eine gültige photometrische Analyse nicht erfüllen.

Wir können auch sehen, dass die PCC drei Koeffizienten gefunden hat, die auf die Farbkanäle anzuwenden sind, um den Weißabgleich zu korrigieren. Die *Abweichung* (deviation), d. h. die durchschnittliche absolute Abweichung der Farbkorrektur für jeden Stern des photometrischen Satzes, ist hier mäßig hoch. Bei gut kalibrierten Bildern ohne Gradient, mit korrekten Filtern und ohne einen Farbnebel, der das gesamte Bild bedeckt, würde die Abweichung näher bei 0 liegen.

Siril Kommandozeile

```
pcc [-limitmag=[+-]] [-catalog=] [-bgtol=lower,upper]
```

Führt die photometrische Farbkalibrierung auf dem geladenen, astrometrisch gelösten Bild durch.

Die Grenzgröße der Sterne wird automatisch aus der Größe des Bildfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.

Der verwendete Sternkatalog ist standardmäßig NOMAD, er kann durch Angabe von **-catalog=apass**, **-catalog=localgaia** oder **-catalog=gaia** geändert werden. Falls NOMAD lokal installiert ist, kann der Remote-NOMAD (vollständige Version) durch Angabe von **-catalog=nomad** erzwungen werden

Die Toleranz für Außreißer der Hintergrundreferenz kann in Sigma-Einheiten mit **-bgtol=lower,upper** angegeben werden: die Standardwerte sind -2,8 und +2,0

10.2.3 Spektrophotometrische Farb-Kalibrierung

Warnung

Die Kalibrierung der Farben durch Photometrie **muss** unbedingt an einem linearen Bild durchgeführt werden, dessen Histogramm noch nicht gestreckt wurde. Andernfalls wird die photometrische Messung falsche Ergebnisse liefern und es gibt keine Garantie für die Korrektheit der erhaltenen Farben.

Spectrophotometric Color Calibration (`Ctrl` + `Shift` + `C`) is the newest method of color calibration available in Siril. This method uses the extensive spectral data available in the Gaia DR3 catalogue [GaiaDR3]. This can be accessed either through direct querying of the online catalogue or by downloading a local extract and querying the local catalogue.

Warnung

Beachten Sie, dass die SPCC-Funktionalität von Siril über den Remote-Katalog nicht verfügbar ist, wenn das Gaia-Archiv aufgrund von Wartungsarbeiten oder einer Störung offline ist. Glücklicherweise ist das Archiv normalerweise sehr zuverlässig, jedoch ist eine Statusanzeige in den SPCC-Dialog integriert. Der Archivstatus wird beim Start des Dialogs überprüft und kann durch Klicken auf die Status-Schaltfläche erneut überprüft werden.

The offline Gaia SPCC extract will still work fine if the Gaia archive is offline.

Tipp

Was ist der Unterschied zwischen SPCC und PCC? Wann soll man die Eine oder die andere Variante benutzen? SPCC ist eine genauere Version der PCC und macht letztere überflüssig. SPCC berücksichtigt den Sensor und die Filter Ihres Equipments. Dadurch ist die erzeugte Farbe viel näher an der "Realität". Das Beispiel in der Abbildung unten veranschaulicht den Unterschied in den Ergebnissen.

Abb. 14: Vergleich zwischen PCC (links) und SPCC (rechts): Zum Vergrößern anklicken. (Mit freundlicher Genehmigung von Ian Cass)

Local SPCC Catalog

From 1.4.0 an offline SPCC catalog is available using Gaia DR3 data. Note that the catalog is chunked into 48 files covering each level 1 HEALpixel.

Theorie

HEALpix (Hierarchical Equal Area isoLatitude Pixelisation) is an algorithm for pixelising a sphere based on subdivision of a distorted rhombic dodecahedron. Mathematical details can be found on Wikipedia [Wiki_HEALPIX]. Gaia sources use a Level 12 NESTED HEALpix scheme and the HEALpixel number is encoded into the source_id. The specification of the Gaia DR3 catalogue extracts and their file format is documented here (PDF).

The nested nature of the scheme means that HEALpixels that are close together in the sky have numbers that are close together. The hierarchical property also means that it is possible to index sources in HEALpixels at a deep HEALpixel level and divide the catalog into chunks at a shallower level while still supporting a highly efficient catalog search algorithm.

It is possible to download the entire catalog or only the chunks you need. The folder location to store the catalog files

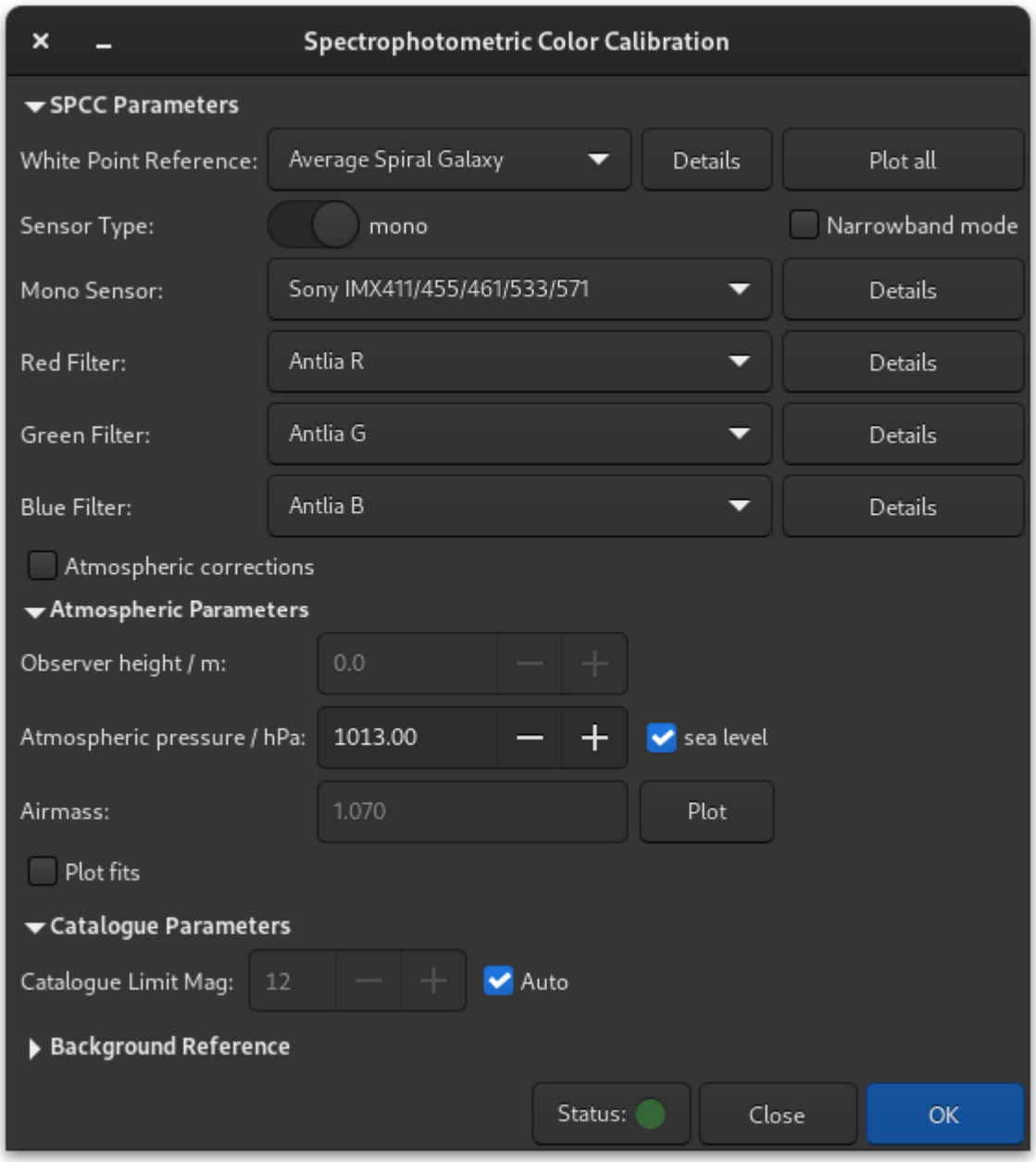
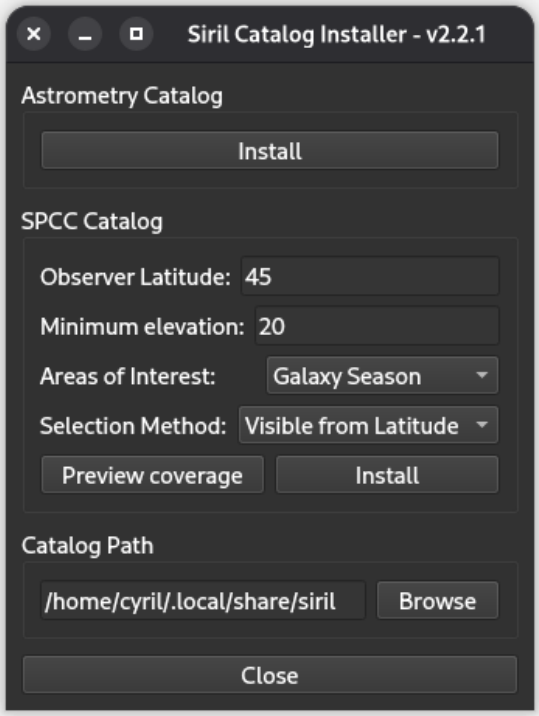


Abb. 13: Dialogfenster Spektro-Photometrische Farbkalibrierung.

is set in *Preferences->Astrometry*.

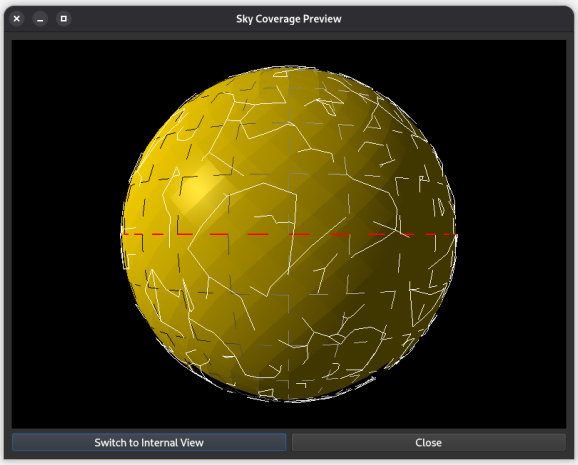
Sirilpy Skript

The easiest way to install the catalog is to use the built-in Python script `Siril_Catalog_Installer.py` in the *Scripts->Python Scripts* menu.



This provides an interface that allows you to install either the whole catalog, or only the chunks that are visible from your observing latitude above a certain elevation, or only sets of chunks corresponding to certain themes (*Milky Way*, *Summer Triangle*, *Galaxy Season* etc.) Select the latitiude / elevation or area of interest if desired, and then select the selection method (*All*, *Visible from Latitude* or *Area of Interest*).

You can preview the coverage using the *Preview coverage* button.



Finally, clicking *Install* will download, verify, uncompress and install the selected chunks and also set the catalog path in Siril's preferences. A default catalog path is suggested in the text entry widget, but can be changed to a different location if you prefer.

If you wish to install the offline SPCC catalog files manually, they can be downloaded from Zenodo. Either individual level 1 HEALpixels can be downloaded or the entire catalog can be downloaded as an archive.

Tipp

When you download "All Files" from the Zenodo record, the download is a zip archive that you will need to extract, however the zip archive is just a convenient way of bundling all of the individual files; the data files inside the zip archive are themselves compressed with bzip2 compression, and you will need to decompress the individual .bz2 files before Siril can use them. Support for this compression format is available by default in Linux and MacOS, and is provided in Windows by various archive programs including 7-Zip and Pea-Zip, which are both Free and Open Source software.

All compressed files have accompanying sha256sums and there is a file containing all the sha256sums of the uncompressed files as well, for additional validation. The Zenodo record also provides a DOI reference that can be used to cite the dataset if you use it in academic work.

Siril uses an optimized extract of the Gaia DR3 xp_sampled datalink product. As with the astrometric extract, the offline catalogue is capped at the 127 brightest sources per level 8 HEALpixel. The catalogue contains fewer sources than the astrometric extract as xp_sampled spectra are typically only provided for sources brighter than magnitude 17.6 and therefore more HEALpixels in emptier parts of the sky have fewer than 127 sources compared with the astrometric extract (i.e. these HEALpixels contain all the available Gaia DR3 sources with xp_sampled data), but this approach still avoids overpopulation of the catalogue in extremely crowded parts of the sky while providing the best SNR. In those HEALpixels with fewer than 127 xp_sampled sources, the local catalog is as comprehensive as using the online Gaia archive directly.

The xp_sampled is converted from float32 to float16 data with an additional byte setting the exponent to be applied to the xp_sampled data for the source to overcome limitations on exponents expressible with float16. This is entirely justifiable given the error bars on the xp_sampled data and makes no practical difference to the accuracy of the results. It means that we can provide a highly effective, purpose-optimized local SPCC catalogue in under 21GB of data.

How it Works

SPCC erfordert Kenntnisse über Ihren Sensor und die von Ihnen verwendeten RGB-Filter. Diese werden über ein *Online-Repository* bereitgestellt, das Siril entweder automatisch beim Start oder bei Bedarf manuell synchronisiert. Sensor- und Filterinformationen werden über dieselbe Synchronisierungsmethode aktualisiert, die auch für das Online-Skript-Repository verwendet wird. (Dies bedeutet, dass Daten zu neuen Filtern oder Sensoren, sobald sie verfügbar sind, dem Repository hinzugefügt werden können, ohne dass eine Aktualisierung der Anwendung erforderlich ist.)

In the GUI you select your sensors and filters from the widgets in the SPCC dialog. Don't worry if there isn't an exact match for your equipment, just pick the closest option, or the appropriate default option. You also need to select a white reference. The default reference is the Average Spiral Galaxy reference which is suitable for a wide range of astrophotographic scenes, however there is an extensive range of galaxy and star types to choose from. The Sun's spectral type is G2(v) so if you want to balance your image using sunlight as a white reference, you would pick **Star, type G2(v)** from the list.

SPCC then uses the stellar spectra in Gaia DR3 and knowledge of your imaging sensor and filters to compute for each star in the catalogue that matches a star detected in the image by Siril the expected flux in each color channel. It then compares this with the actual flux measured in each channel using Siril's photometric capabilities.

Given the sensor and filter knowledge, SPCC computes the expected flux in each channel for the specified white reference. A robust linear fit is obtained to give the best fit of catalogue to image R/G and B/G flux ratios for each star and for the white reference. This fit is used to derive correction coefficients which are applied multiplicatively to each channel, resulting in spectrophotometrically accurate color channels.

Your image must be plate solved for SPCC to work: if it is not already, this should be done with the dedicated tool. It is important to make sure that the plate solving information is correct, as some software is known to add inaccurate WCS data to images.

Grafische Benutzeroberfläche

- **Selection of Sensor** In order to select your sensor, ensure that the mono / OSC toggle button is set correctly. You will then see the appropriate dropdown to choose from the available sensors.
- **Selection of Filters** SPCC can operate in two modes.
  - The default mode is broadband operation. In this mode, the *Narrowband mode* check box should be unchecked. You can choose either red, green and blue filters (for composited images made with a mono sensor) or OSC filters, for example light pollution filters, for images made with an OSC sensor.

    > **Warnung**
    >
    > If you select a DSLR OSC sensor (e.g. a Canon EOS 600D) an additional widget will become visible to select a DSLR Low Pass Filter. This allows you to tailor whether your camera has been astro-modded or not. You **must** select an option here or the process will complain that you haven't set all the necessary filters!
    >
    > Options exist for Canon and Nikon OEM low-pass filters as well as the popular Baader BCF astro-mod filter that lets Ha and Sii through but stil blocks longer IR wavelengths and "Full spectrum" which is modelled as a perfect clear filter.
    >
    > If you have an unmodified camera of a different model or brand, select any of the Canon or Nikon low-pass filters: the effect is very minor as these wavelengths are right at the edge of human visual perception anyway.

  - By checking the *Narrowband mode* check box, you enable narrowband mode. This is intended either for images composited from narrowband filters used with a mono sensor or for images made using an OSC sensor with a dual, tri-band or quad band narrowband filter. In this mode the available controls change, and for each color channel you enter the nominal wavelength and bandwidth of the filter passband. For

ultra-narrowband mono filters the passband may be as little as 3nm; for a quadband OSC filter like the Altair QuadBand V2 the passbands may be as much as 35nm. Note that for a HOO composition where two channels are set to the same data, the nominal wavelength and bandwidth should be set equal in the SPCC interface too.

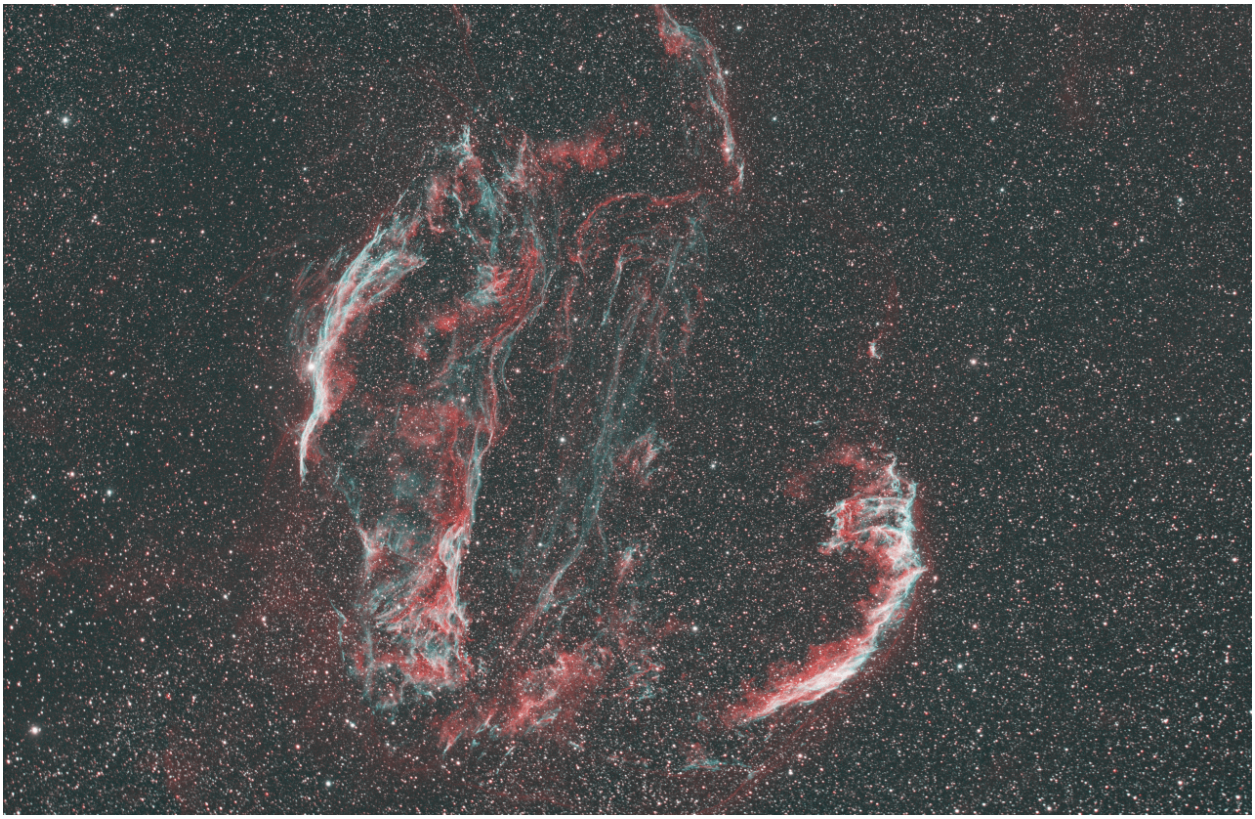


Abb. 15: Kalibriertes HOO-Bild (Bildquelle: Cyril Richard).

Tipp

Some manufacturers specify a center wavelength and FWHM. It is fine to use the FWHM as the bandwidth: these filters have very sharp cutoffs.

Warnung

Erwarten Sie nicht, dass Sie die Hubble-Palette für SHO-Abbildungen mit den Wellenlängen der Filter SII, Hα und OIII abrufen können. Das Ergebnis wird ein Bild mit einem starken Grünstich sein. Dies lässt sich leicht dadurch erklären, dass die SII-Emissionslinie viel schwächer ist als die des Wasserstoffs und das SPCC eine Darstellung der tatsächlichen Intensitäten liefert. Dies ist bei der Hubble-Palette jedoch nicht der Fall. Mit der *manuellen Farbkalibrierung* werden bessere Ergebnisse erzielt.

- **Selection of DSLR Low Pass Filter (LPF)** DSLRs contain a low-pass filter (sometimes also called a 'hot mirror'. These reduce transmittance at wavelengths of interest to astronomers (Ha at 656nm and S-II at 674nm). If the selected OSC is a DSLR, a dropdown will be provided from which you can the appropriate LPF profile. Options exist for stock LPFs as well as astro-modified LPFs and an ideal **Full spectrum** filter model for if the LPF has been removed altogether.

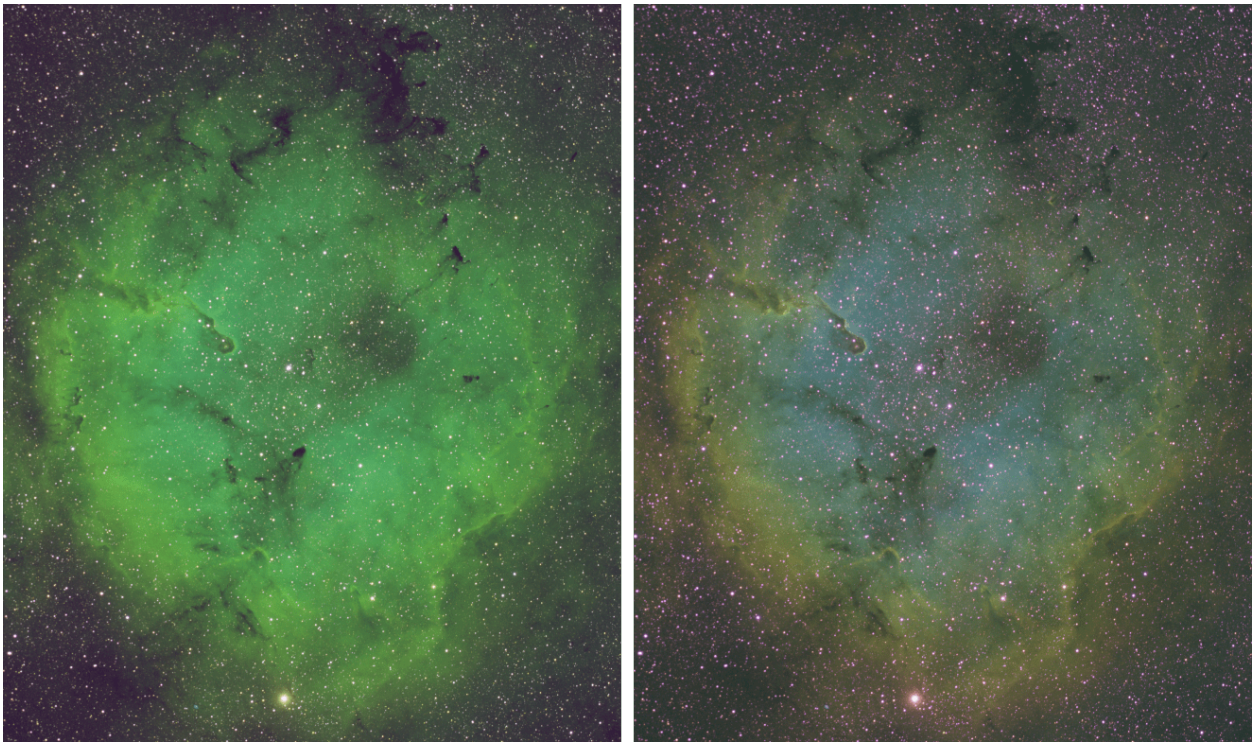


Abb. 16: SHO image calibrated by SPCC compared to the same, manually calibrated one. The entire nebula was taken as a white reference during manual calibration. Image by Cyril Richard.

- **Selection of White Reference** SPCC requires an absolute white reference spectrum. The default is **Average Spiral Galaxy** and the source spectra used to create this white reference are taken from the SWIRE templates [SWIRE] in a manner consistent with other astrophotography software providing the same white reference. A wide range of other white references is available, covering the full range of galaxy and star classifications [Stellar]. If you wish to use sunlight as your white reference, you would choose the white reference **Star, type G2(v)** as the Sun is a type G2(v) star.

> **Tipp**
>
> **Summary of Stellar Spectral Classifications** Stellar classifications have two parts, a Morgan-Keenan type and a Luminosity index.
>
> The first part of the spectral classification (G2 in the case of the Sun) takes one of the following letters: O, B, A, F, G, K, M. O represents extremely hot blue stars, while M represents cool red stars. The sun is roughly in the middle of the spectrum. The number represents intermediate cases, for example a B5 star is halfway between type B and type A.
>
> The second part of the spectral classification is the luminosity index ranging from i to v. Stars with luminosity index i are supergiants, whereas stars with luminosity index v are dwarfs. Main sequence stars such as the sun have a luminosity index of iv.

- **Atmospheric Correction** Siril's SPCC supports atmospheric correction. When imaging from Earth we image through the atmosphere. This does not have perfect transmittance and therefore acts as another, non-optional, filter in the imaging chain between the sensor and the astronomical object. Whether or not to correct for this is an artistic choice each astrophotographer must make, but the option is provided.

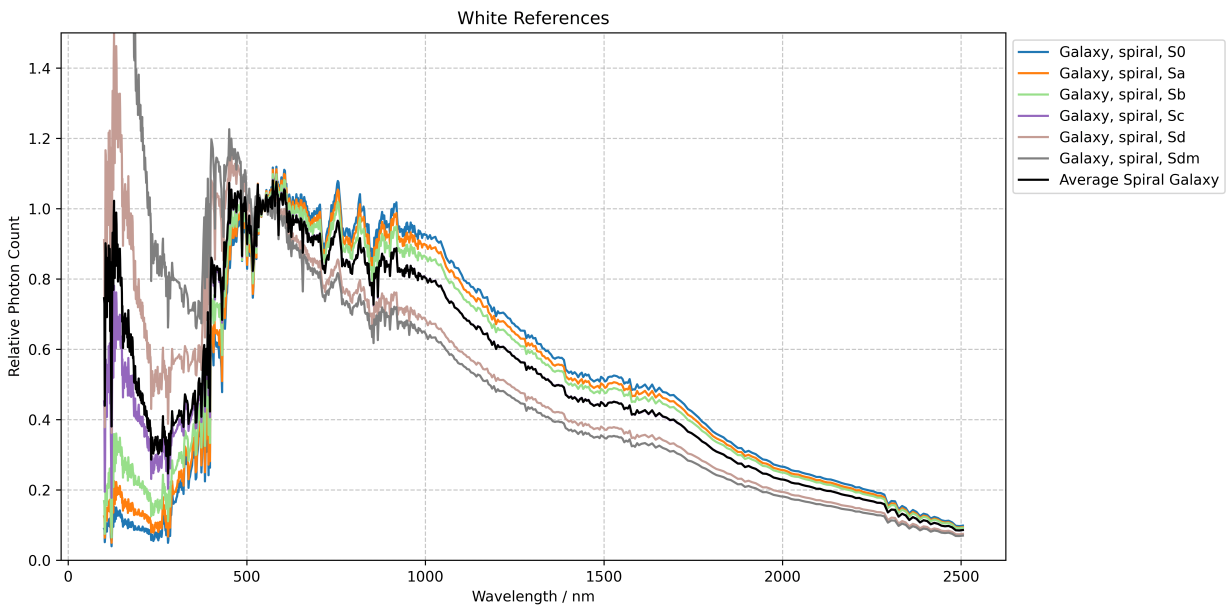


Abb. 17: Graphs showing white reference data from spiral galaxies. At around 350 nm, the Average Spiral Galaxy data become identical to the Sc galaxies, which are also a good representation of the white reference.

Theorie

Atmospheric extinction arises from several sources. The most important are:

- Rayleigh scattering. This is the elastic scattering of light by particles that are small compared with the wavelength of light. The strong wavelength dependence of the Rayleigh scattering ($\approx^4$) means that shorter (blue) wavelengths are scattered more strongly than longer (red) wavelengths.
- Aerosol scattering. This is scattering of light by particles that are larger than the wavelength of light. This is quite variable but (in the absence of significant short term dust or smoke effects) relatively spectrally flat and less significant than Rayleigh scattering.
- Molecular absorption lines.

Siril models only Rayleigh scattering. This is the most important contribution in most atmospheric conditions, and is highly predictable making it easy to model without requiring the user to provide complex data.

The formula for the Rayleigh transmittance of the atmosphere as a function of wavelength λ nm, observer height H m and pressure p hPa is:

$$\tau_R(\lambda, H, p) = \left(\frac{p}{1013.25}\right)\left(0.00864 + 6.5 \times 10^{-6} \cdot H\right)\lambda^{-(3.916+0.074\lambda+\frac{0.050}{\lambda})}.$$

Under normal circumstances aerosol scattering has a roughly flat response in the visible region. This changes in specific conditions, for example when there is high atmospheric smoke particle concentration after wildfires or, in parts of Europe, when Saharan dust is carried into the atmosphere. However these effects are very difficult to model accurately as they depend on the concentration of sand or smoke particles in the atmosphere at the time. Siril therefore does not model this effect.

The main molecular absorption lines in the visible spectrum are the Chappuis stratospheric ozone bands and the Fraunhofer B molecular oxygen absorption line. However the Fraunhofer B line is very narrow and does not have a significant effect on overall calibration. The Chappuis bands are broad but with a low peak absorption, with a much smaller overall impact than Rayleigh scattering. Molecular absorption bands are not currently modelled in Siril.

Abb. 18: NGC 4414 is a great example of a Sc-type galaxy, the type closest of the average spiral galaxy used as white reference by default. Image Credit: NASA, ESA, W. Freedman (U. Chicago) et al, & the Hubble Heritage Team (AURA/STScI), SDSS; Processing: Judy Schmidt.

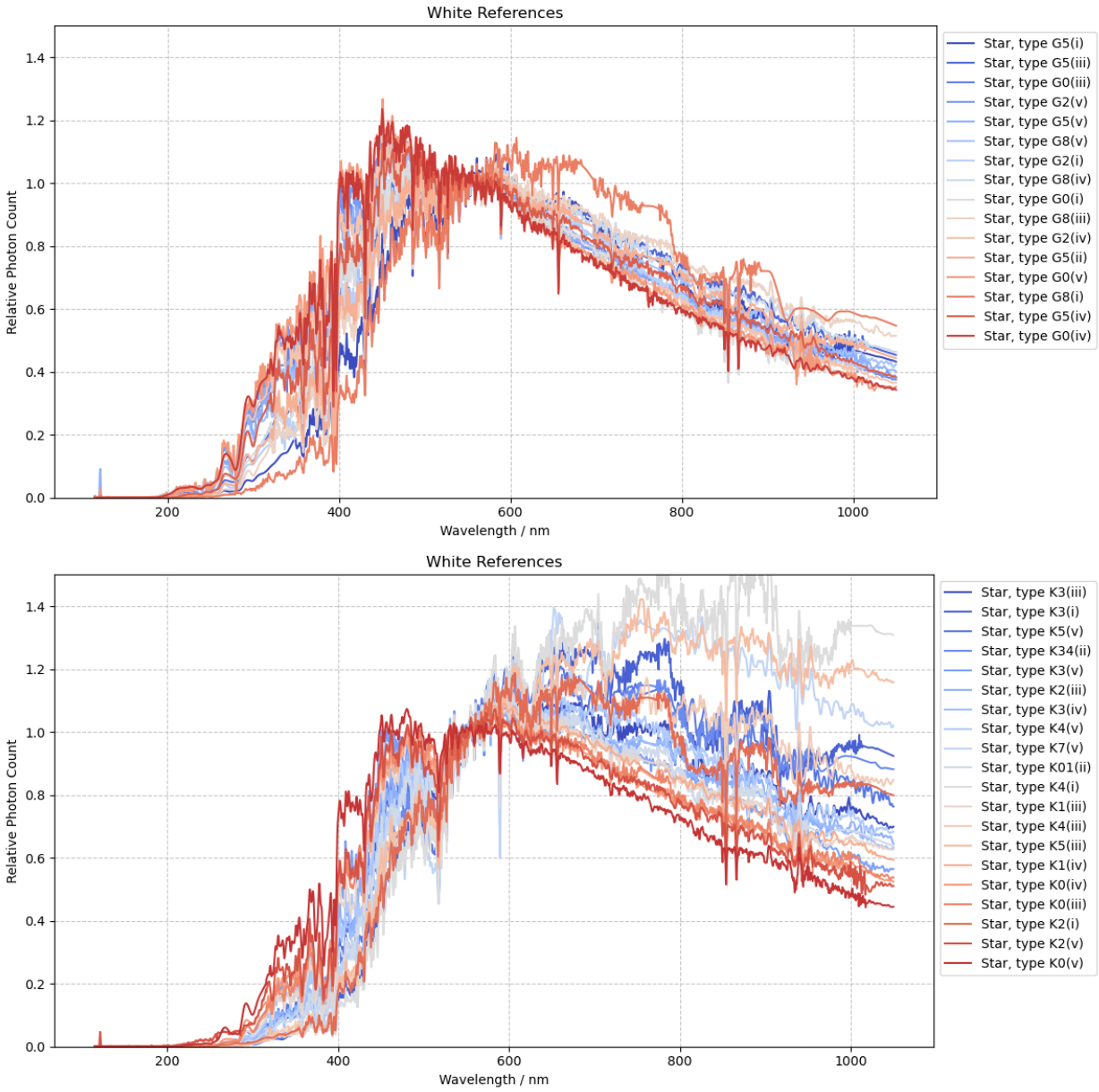


Abb. 19: Graphs showing white reference data for a set of two different star classes, G and K.

Abb. 20: Difference in color calibration depending on the choice of white reference. On the left, an M-type star, on the right the average spiral galaxy. Please note that the data are linear, and only an autostretch has been applied to the visualization.

When selecting the *Atmospheric correction* check box, the following options become available:

- Observer Height. This allows setting of the observer height, which is used in the Rayleigh extinction calculation. Set this to the altitude of your observatory above sea level. Some capture software sets the FITS header `SITEELEV` card: if this is present, the height from this card will be used, otherwise the value is editable and defaults to 10 m.
- Atmospheric pressure. This allows setting atmospheric pressure at the time of observation. For convenience it can be specified as sea level pressure (as provided by weather forecasts) or as local pressure (as measured by a barometer at the observatory). In case you are unsure, the default is standard atmospheric pressure at sea level (1013.25 hPa).

Theorie

If the pressure is provided as a sea-level pressure measurement, the local pressure at the observer's height is calculated according to the barometric formula:

$$P(h) = P_0 \left(1 - \frac{Lh}{T_0}\right)^{\frac{gM}{RL}},$$

wobei:

* $L = 0.0065$ K/m (Temperature lapse rate),
* $T_0 = 288.15$ K (Sea level standard temperature),
* $g = 9.80665$ m/s² (Acceleration due to gravity),
* $M = 0.0289644$ kg/mol (Molar mass of Earth's air),
* $R = 8.3144598$ J/(molK) (Universal gas constant).

- Airmass. This is not an editable parameter but shows the airmass that will be used in the calculations. It is obtained, in order of preference, from the `AIRMASS` FITS header card; by calculation using the `CENTALT`

FITS header card; or as a last resort by using the average zenith angle of all parts of the more than 30° above the horizon. The tooltip shows which source the used figure is based on.

Theorie

If the `AIRMASS` header is unavailable the calculation used to derive airmass from zenith angle is calculated in accordance with [Young1994]:

$X(z) = \frac{1.002432\cos^2 z + 0.148386\cos z + 0.0096467}{\cos^3 z + 0.149864\cos^2 z + 0.0102963\cos z + 0.000303978}$.

The interface allows you to view details of the selected sensor, filter and white reference using the *Details* button next to each combo box. From the details information box that this brings up you also have the option to plot the Quantum Efficiency (for sensors) or transmittance (for filters) or relative photon count (for white references) against wavelength. A *Plot All* button is also available in the main SPCC dialog which allows you to see the responses of all your filters and your sensor and the white reference spectrum all plotted together.

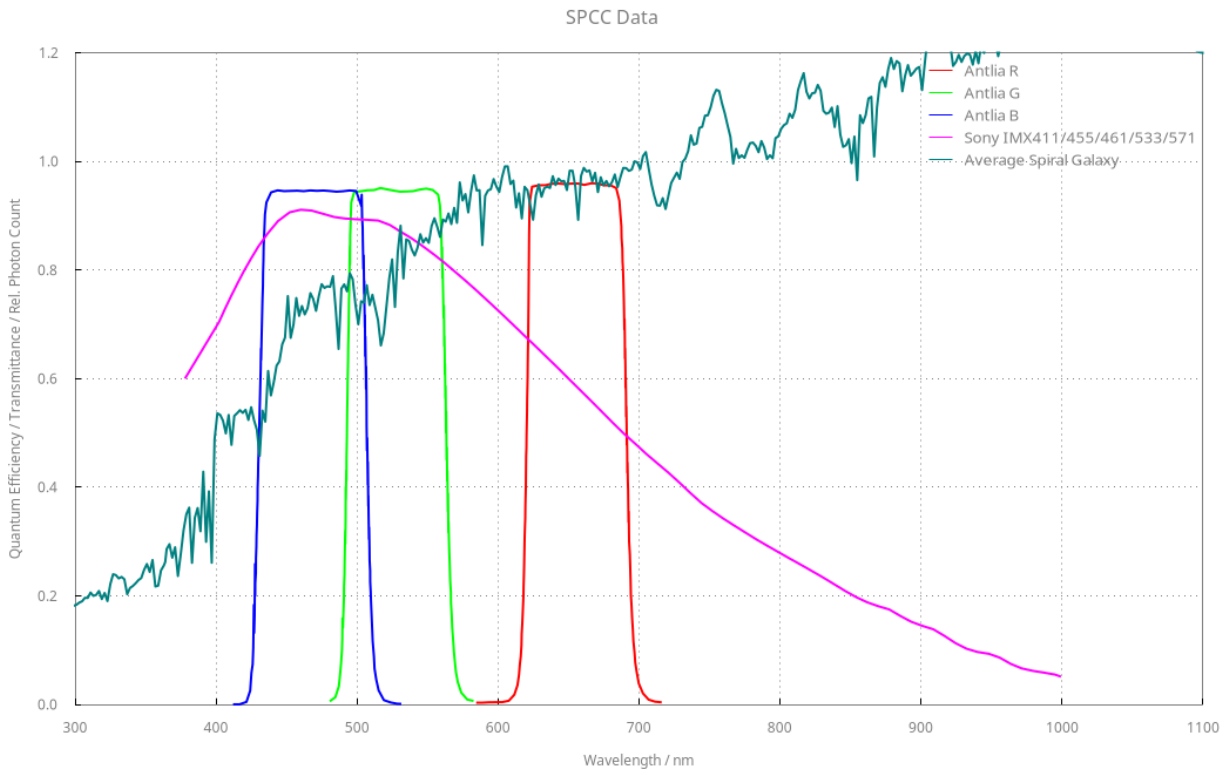


Abb. 21: Plotting all the responses of all your filters and your sensor and the white reference spectrum all plotted together

When you are happy, click *Apply* and SPCC will run. It will cache catalogue data but the first time you apply it to an image it will take a few seconds to perform the online catalogue searches and retrieve the source and spectral data. SPCC will then be applied to the image. Additional plots showing the linear fit of the catalogue Red / Green and Blue / Green to image Red / Green and Blue / Green ratios.

Tipp

How do I process L-RGB images? We recommend processing only RGB with SPCC. The L layer must be added

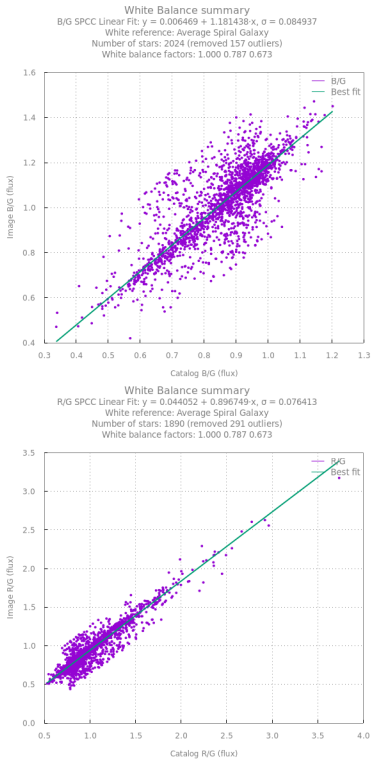


Abb. 22: By default, Siril outputs graphs showing the fits used in the process. In this example the magnitude was limited to 17.

at a later stage, when the histograms have been stretched.

Tipp

Für Bilder, die mit einem OSC-Sensor aufgenommen wurden, empfehlen wir die Verwendung von *Bayer Drizzle* zur Wiederherstellung der Bildfarben. Dies gewährleistet genauere Farben, wie im folgenden Bild gezeigt wird.

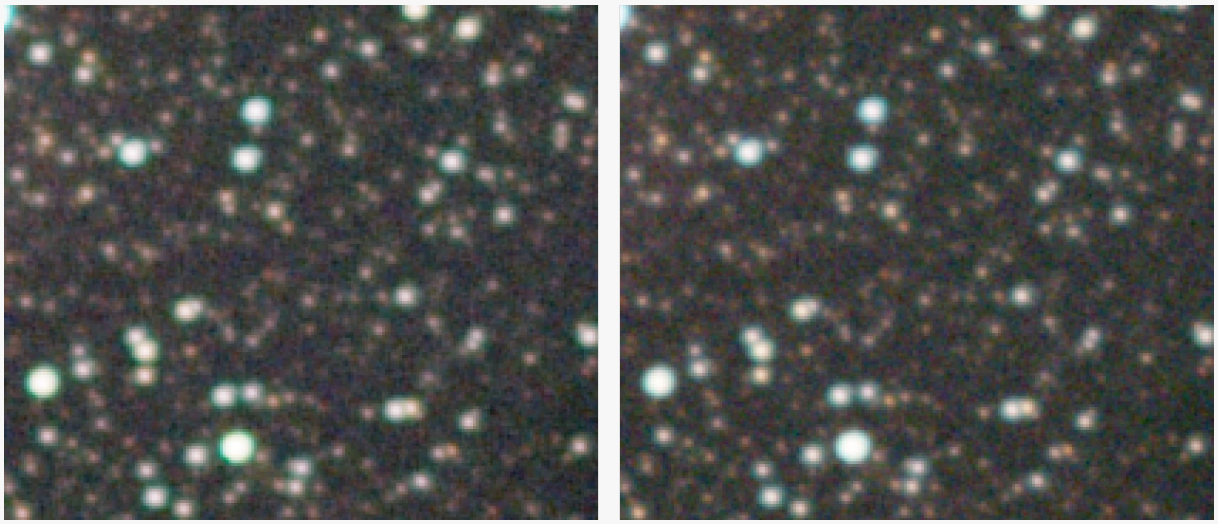


Abb. 23: SPCC in identischer Weise auf dasselbe Bild angewendet. Links: konventionelles Demosaicing mit dem VNG-Algorithmus; rechts: Bayer-Drizzle-Technik. Auf dem konventionell debayerten Bild ist ein dominanter Grünton deutlich sichtbar. **Der VNG-Algorithmus wurde für dieses Beispiel gewählt, weil die hier erläuterten Effekte stärker ausgeprägt sind. In Siril ist der Standard-Demosaikierungsalgorithmus jedoch RCD. Klicken Sie zum Vergrößern auf das Bild.

SPCC-Filter- und Sensordatenbank

Daten konvertieren

Für die Datenbank wird das JSON-Format (ein schlankes Datenaustauschformat abgeleitet von der Notation für JavaScript-Objekte) benutzt. Wir empfehlen, mit einer vorhandenen Datei aus der Datenbank zu beginnen, die Ihren Anforderungen entspricht, und diese unter dem Namen Ihres Sensors oder Filters zu speichern. Sie können dann einfach die Werte in der Datei durch die von Ihnen erhaltenen Daten ersetzen.

- Geben Sie im Feld `Wellenlänge` Ihre Wellenlängenmessungen ein. Stellen Sie sicher, dass das Feld `Einheiten` auf einen der folgenden Werte eingestellt ist: `Angström`, `nm`, `Mikrometer` oder `m`.
- Geben Sie im Array `Werte` Folgendes ein:
  - Durchlasswerte für Filter
  - Quanteneffizienzwerte für Sensoren

  Legen Sie das Feld `Bereich` entsprechend Ihrer Datenskala fest (z.B. `Bereich`: 100, wenn es sich bei Ihren Werten um Prozentwerte handelt, `Bereich`: 1, wenn sie auf 1 normalisiert sind).

Wie man selbst beitragen kann

Die SPCC-Datenbank dient zum Speichern von JSON-Dateien von OSC/monochromen Sensoren und Filtern, die auf dem Markt erhältlich sind. Ihr Hauptziel ist das Sammeln umfangreicher Daten und die Förderung der Zusammenarbeit innerhalb der Community.

Wir legen großen Wert auf Beiträge aus der Community und ermutigen zur aktiven Teilnahme. Wir benötigen Daten, die idealerweise von 300 nm bis 1100 nm reichen. Mithilfe von Softwaretools können Kurven/Diagramme aus dem Internet extrahiert werden. Sie können sich auch direkt an die Hersteller wenden, um Daten anzufordern.

Read this page and help us by contributing.

Wichtig

Wir schließen keine Schmalbandfilter ein. Diese hochspezifischen Filter werden in Siril synthetisiert, was Präzision gewährleistet. Dies gilt auch für Duo-Schmalbandfilter.

Formatreferenz für das JSON-Dateiformat

Hier ist die Vorlage für die in der SPCC-Datenbank verwendeten JSON-Dateien:

```
[
  {
    "model": "sensor model / filter set",
    "name": "sensor / filter name",
    "type": "MONO_SENSOR | OSC_SENSOR | MONO_FILTER | OSC_FILTER | OSC_LPF | WB_REF",
    "dataQualityMarker": 1 - 5,
    "dataSource": "Describe where the data came from",
    "manufacturer": "Manufacturer name",
    "version": 1,
    "channel": "RED | GREEN | BLUE | LUM",
    "wavelength": [Comma separated array of wavelengths],
    "values": [Comma separated array of values]
  }
]
```

Wichtige Hinweise

- Definition des Feldes `dataQualityMarker`:
  1. Daten unbekannter Herkunft. **Nicht akzeptiert** für das Siril-SPCC-Datenbank-Repository.
  2. Von OEMs oder anderen namhaften Anbietern im Bildformat gescannte Daten.
  3. Tabellarische Daten mit niedrigerer Auflösung, die vom OEM bereitgestellt werden, oder akademische Daten in Bezug auf die ideale Standardfilterdurchlässigkeit (z. B. generische Standard-Photometriefilter).
  4. Tabellarische Daten mit hoher Auflösung (nicht mehr als 2nm Abstand), vom OEM bereitgestellt.
  5. Daten, die spezifisch für Ihren eigenen Filter sind, den Sie persönlich mit geeigneter Ausrüstung kalibriert haben. Dies ist der höchstmögliche Qualitätsmarker und wird niemals an .json-Dateien im Repository vergeben, die immer nur allgemein für ein Ausrüstungsmodell und nicht spezifisch für Ihr individuelles Ausrüstungselement sein können. Beachten Sie, dass die tatsächliche Qualität dieser Daten vollständig von der Qualität Ihrer Kalibrierungsausrüstung abhängt – das alte Sprichwort „Müll rein, Müll raus“ gilt.
- Die Anforderungen an den Modellnamen `model`:
  - Muss für alle zugehörigen JSON-Objekte in einem Set identisch sein
  - Beispiele:
    * RGB Filtersatz: `"model": "Chroma RGB"`
    * OSC Sensor: `"model": "ZWO ASI2600MM"`
- Das Feld Kanal `channel`:
  - Nur erforderlich für `"type": "OSC_SENSOR"` oder `"type": "MONO_FILTER"`
  - Für OSC-Sensoren fügen Sie ein JSON-Objekt pro Kanal ein (`RED`, `GREEN`, `BLUE`)
  - Bevorzugte Reihenfolge der Kanäle: `RED`, `GREEN`, `BLUE`

- Die Anforderungen des Array `wavelength` (Wellenlänge):
  - Minimale Abdeckung: 380nm bis 700nm
  - Maximal sinnvoller Bereich: 336nm bis 1020nm (Gais DR3 Spektralgrenzen)
  - Die Werte müssen monoton steigend sein
  - Doppelte Werte sind nicht erlaubt
  - Es müssen bestimmte Einheiten verwendet werden (`angstroms`, `nm`, `micrometres`, `m`).

> **Bemerkung**
>
> Wenn Ihre Sensordaten nur bis 400nm reichen (was bei einigen Herstellern üblich ist), ist es akzeptabel, einen einzelnen Punkt bei 380nm zu extrapolieren. Die Sensorreaktion unter 400nm folgt bei verschiedenen Sensoren normalerweise einem vorhersehbaren Muster. Das Hinzufügen dieses extrapolierten Punkts bei 380nm ist besser, als die Kurve bei 400nm enden zu lassen, was die Empfindlichkeit unter 400nm effektiv als Null behandeln würde. Die Auswirkung dieser Extrapolation ist minimal, da die CIE 1931-Empfindlichkeit in diesem Wellenlängenbereich sehr gering ist.

- Die Anforderungen des Arrays `values` (Werte):
  - Für Filter: enthält Transmissionswerte
  - Für Sensoren: enthält Quanteneffizienzwerte
  - Legen Sie einen geeigneten `Bereichswert`, `range` fest (z. B. 100 für Prozentsätze).
  - Siril skaliert alle Werte intern auf einen Bereich von 0,0 - 1,0.

Gespeicherter Einstellungen

As most users are likely to do most of their imaging with one setup, or maybe two, it would be tedious to reselect the sensor and filters each time. The user choices are therefore automatically remembered when set and restored next time the tool is used, even if Siril is closed and restarted in between. This works using the preferences system but there is no need to use the preferences dialog to remember the set sensor and filters, it is done automatically.

The chosen white reference is not remembered: the default Average Spiral Galaxy is a suitable choice for most astronomical scenes, and alternative white references would normally be set for a specific image to draw out a particular aspect of the color.

> **Siril Kommandozeile**
>
> ```
> spcc [-limitmag=[+-]] [ { -monosensor= [ -rfilter= ] [-gfilter=] [-bfilter=] | -
> ↪oscsensor= [-oscfilter=] [-osclpf=] } ] [-whiteref=] [ -narrowband [-rwl=] [-gwl=]␣
> ↪[-bwl=] [-rbw=] [-gbw=] [-bbw=] ] [-bgtol=lower,upper] [ -atmos [-obsheight=] { [-
> ↪pressure=] | [-slp=] } ]
> ```
>
> Führt die spektrophotometrische Farbkalibrierung auf dem geladenen, astrometrisch gelösten Bild durch.
>
> Die Grenzgröße der Sterne wird automatisch aus der Größe des Bildfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.
> The star catalog used for SPCC is always Gaia DR3: by default the local Gaia DR3 xp_sampled catalog will be used if available but this can be overridden with **-catalog={gaia | localgaia}**.

Die Namen der Sensoren und Filter können mit den folgenden Optionen angegeben werden: **-monosensor=**, **-rfilter=**, **-gfilter=**, **-bfilter=** oder **-oscsensor=**, **-oscfilter=**, **-osclpf=**; der Name der Weißreferenz kann mit der Option **-whiteref=** angegeben werden. In allen Fällen muss der Name genau so angegeben werden, wie er in den Kombinationsfeldern des SPCC-Tools steht. Beachten Sie, dass Sensor-, Filter- und Weißreferenznamen Leerzeichen enthalten können: Wenn Sie sie als Argumente für den Befehl **spcc** verwenden, muss das gesamte Argument in Anführungszeichen gesetzt werden, zum Beispiel "-whiteref=Average Spiral Galaxy".

Der Schmalbandmodus kann mit dem Argument **-narrowband** ausgewählt werden. In diesem Fall werden die vorherigen Filterargumente ignoriert, und die Wellenlängen und Bandbreiten der NB-Filter können mit **-rwl=**, **-rbw=**, **-gwl=**, **-gbw=**, **-bwl=** und **-bbw=** angegeben werden.

Wird eines der Spektraldaten-Argumente weggelassen, wird der zuvor verwendete Wert verwendet.

Die Toleranz für Außreißer der Hintergrundreferenz kann in Sigma-Einheiten mit **-bgtol=lower,upper** angegeben werden: die Standardwerte sind -2,8 und +2,0

Eine atmosphärische Korrektur kann durch Übergabe von **-atmos** vorgenommen werden. In diesem Fall gelten die folgenden optionalen Argumente: **-obsheight=** gibt die Höhe des Beobachters über dem Meeresspiegel in Metern an (Standard 10), **-pressure=** gibt den lokalen Luftdruck am Beobachtungsort in hPa an oder **-slp=** gibt den Luftdruck auf Meereshöhe in hPa an (der Standarddruck beträgt 1013,25 hPa auf Meereshöhe).

```
spcc_list { oscsensor | monosensor | redfilter | greenfilter | bluefilter | oscfilter␣
↪| osclpf | whiteref }
```

Zeigt eine Liste von SPCC-Namen an, die zur Definition von Sensoren, Filtern oder Weißreferenzen mit dem Befehl **spcc** verwendet werden können. Dieser Befehl erfordert ein Argument, um festzulegen, welche Liste gedruckt werden soll: Die Optionen sind **oscsensor**, **monosensor**, **redfilter**, **greenfilter**, **bluefilter**, **oscfilter**, **osclpf** oder **whiteref**.

Beachten Sie, dass Sensor-, Filter- und Weißreferenznamen Leerzeichen enthalten können: Wenn Sie sie als Argumente für den Befehl **spcc** verwenden, muss das gesamte Argument in Anführungszeichen gesetzt werden, zum Beispiel "-whiteref=Average Spiral Galaxy"

Verweise: *spcc*

Quellenverzeichnis

10.3 Andere Farbwerkzeuge

10.3.1 Farbsättigung

Dieses Werkzeug wird verwendet, um die Farbsättigung des Bildes zu verändern. Sie können zwischen einem bestimmten Farbton oder dem Gesamtfarbton (Global) wählen, der verstärkt oder vermindert werden soll. Die Stärke der Sättigung wird mit dem Schieberegler *Stärke* eingestellt.

Mit dem Schieberegler *Hintergrundfaktor* wird der mit dem Hintergrundwert multiplizierte Faktor eingestellt. Je niedriger der Wert, desto stärker ist der Sättigungseffekt. Bei einem hohen Wert bleibt der Hintergrund erhalten.

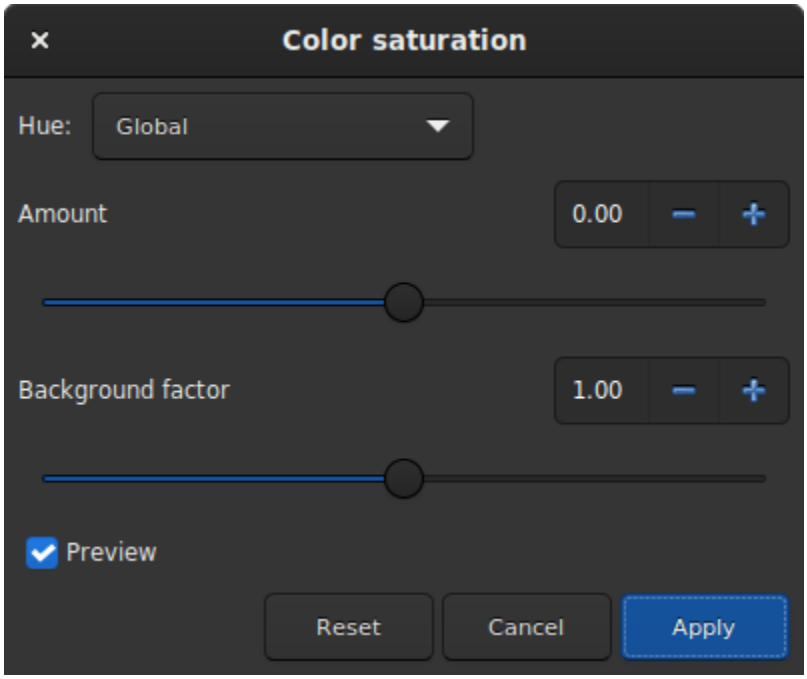


Abb. 24: Dialogfenster Farbsättigung.

Siril Kommandozeile

```
satu amount [background_factor [hue_range_index]]
```

Erhöht die Farbsättigung des geladenen Bildes. Versuchen Sie es iterativ, um beste Ergebnisse zu erzielen.

amount kann eine positive Zahl sein, um die Farbsättigung zu erhöhen, eine negative, um sie zu verringern, 0 würde nichts bewirken, 1 würde sie um 100% erhöhen

background_factor ist ein Faktor (Median + Sigma), der verwendet wird, um einen Schwellenwert festzulegen, bei dem nur die Pixel oberhalb dieses Wertes geändert werden. Dies ermöglicht es, dass das Hintergrundrauschen nicht farblich gesättigt wird, wenn es sorgfältig gewählt wird. Der Standardwert ist 1, mit 0 wird der Schwellenwert deaktiviert.

hue_range_index kann [0, 6] sein, was bedeutet: 0 für rosa bis orange, 1 für orange bis gelb, 2 für gelb bis cyan, 3 für cyan, 4 für cyan bis magenta, 5 für magenta bis rosa, 6 für alle (Standard)

10.3.2 Grün-Rauschen entfernen

Da Grün in Deep-Sky-Bildern von Natur aus nicht vorkommt (mit Ausnahme von Kometen und einigen planetarischen Nebeln), können wir davon ausgehen, dass Grün zum Rauschen gehört, wenn das Bild bereits kalibriert wurde, die Farben ausgewogen sind und das Bild frei von jeglichen Farbverläufen ist. Es ist dann interessant, eine Methode zu finden, um dieses dominante Grün zu entfernen. Genau das schlägt das Werkzeug Grünes Rauschen entfernen vor, das vom Werkzeug Subtraktive Farbrauschunterdrückung abgeleitet ist, aber nur für Grün.

Warnung

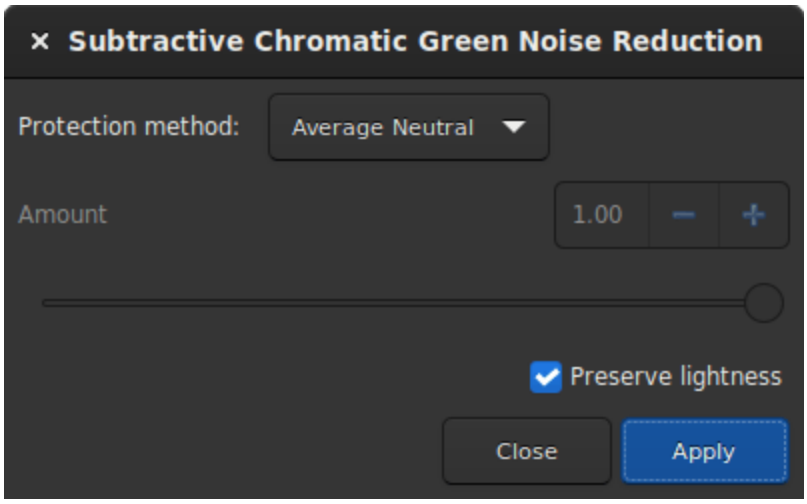


Abb. 25: Dialogfenster Grün-Rauschen entfernen.

Dieses Werkzeug ist nicht für die direkte Verwendung auf einem typischen grünen Bild aus dem Stacking vorgesehen, bei dem der Pegel des Hintergrunds nicht ausgeglichen wurde. Seine Verwendung unter solchen Bedingungen würde die Farbigkeit des Bildes zerstören.

Warnung

Dieses Werkzeug ist für die Verwendung mit nichtlinearen Bildern konzipiert, da es in einem nichtlinearen Farbraum arbeitet. Stellen Sie sicher, dass das Histogramm gestreckt wurde, bevor Sie dieses Werkzeug verwenden, um falsche Verarbeitungsergebnisse zu vermeiden.

Dieses Werkzeug hat 3 Einstellungen. Die Schutzmethode, den Betrag (im folgenden Abschnitt a genannt) und eine Schaltfläche *Erhalte Helligkeit*. Die folgenden Methoden zeigen die verschiedenen Möglichkeiten, die grünen Pixel durch eine Mischung aus Rot und Blau zu ersetzen. Der Wert *Stärke* ist nur für Methoden mit Maskenschutz verfügbar. Der Wert muss mit Bedacht gewählt werden, um den Anstieg des Magentastichs im Himmelshintergrund zu minimieren. Zögern Sie nicht, die Schaltflächen *Undo* und *Redo* zu verwenden, um den Wert fein abzustimmen.

Schutzmethode

Maximaler Maskenschutz

$$m = \max(R, B)$$
$$G' = G \times (1^{-}a) \times (1^{-}m) + m \times G$$

Additiver Maskenschutz

$$m = \min(1, R + B)$$
$$G' = G \times (1^{-}a) \times (1^{-}m) + m \times G$$

Durchschnittlicher neutraler Maskenschutz (Standardmethode)

$$m = 0.5 \times (R + B)$$
$$G' = \min(G, m)$$

Maximaler Neutralschutz

$$m = \max(R, B)$$
$$G' = \min(G, m)$$

Die Schaltfläche *Erhalte Helligkeit* schließlich bewahrt die ursprüngliche CIE L*-Komponente im verarbeiteten Bild. Um nur die chromatische Komponente zu verarbeiten, ist es sehr empfehlenswert, diese Option aktiviert zu lassen.

Siril Kommandozeile

```
rmgreen [-nopreserve] [type] [amount]
```

Wendet einen Filter zur Reduzierung des chromatischen Rauschens an. Er entfernt den Grünstich im aktuellen Bild. Dieser Filter basiert auf dem SCNR-Filter von PixInsight und ist auch der gleiche Filter, der vom HLVG-Plugin in Photoshop verwendet wird.

Die Helligkeit bleibt standardmäßig erhalten, kann aber mit dem Schalter **-nopreserve** deaktiviert werden.

Typ kann die Werte 0 für durchschnittliche Neutralität, 1 für maximale Neutralität, 2 für maximale Maske, 3 für additive Maske annehmen, wobei der Standardwert 0 ist. Die letzten beiden können ein Argument **Betrag** annehmen, einen Wert zwischen 0 und 1, wobei der Standardwert 1 ist

10.3.3 Negative Transformation

Negative Transformation bezieht sich auf das Subtrahieren von Pixelwerten von $(L-1)$, wobei L der maximal mögliche Wert des Pixels ist, und das Ersetzen durch das Ergebnis.

Das Werkzeug *Negativtransformation* unterscheidet sich von der Negativansicht in der Werkzeugleiste. Die Transformation ist nämlich nicht nur visuell, sondern wird tatsächlich auf die Pixelwerte angewendet. Wenn Sie das Bild speichern, wird es als Negativ gespeichert.

Tipp

Eine häufige Anwendung der Negativtransformation ist die Entfernung des Magentastichs aus SHO-Bildern. In diesem Fall muss man *Negative Transformation* anwenden, dann *Grün- Rauschen entfernen*, dann wieder *Negative Transformation*.

Siril Kommandozeile

```
neg
```

Ändert die Pixelwerte des aktuell geladenen Bildes in eine negative Ansicht. Beispielsweise "1-value" für 32-Bit, "65535-value" für 16-Bit. Dies ändert nicht den Anzeigemodus

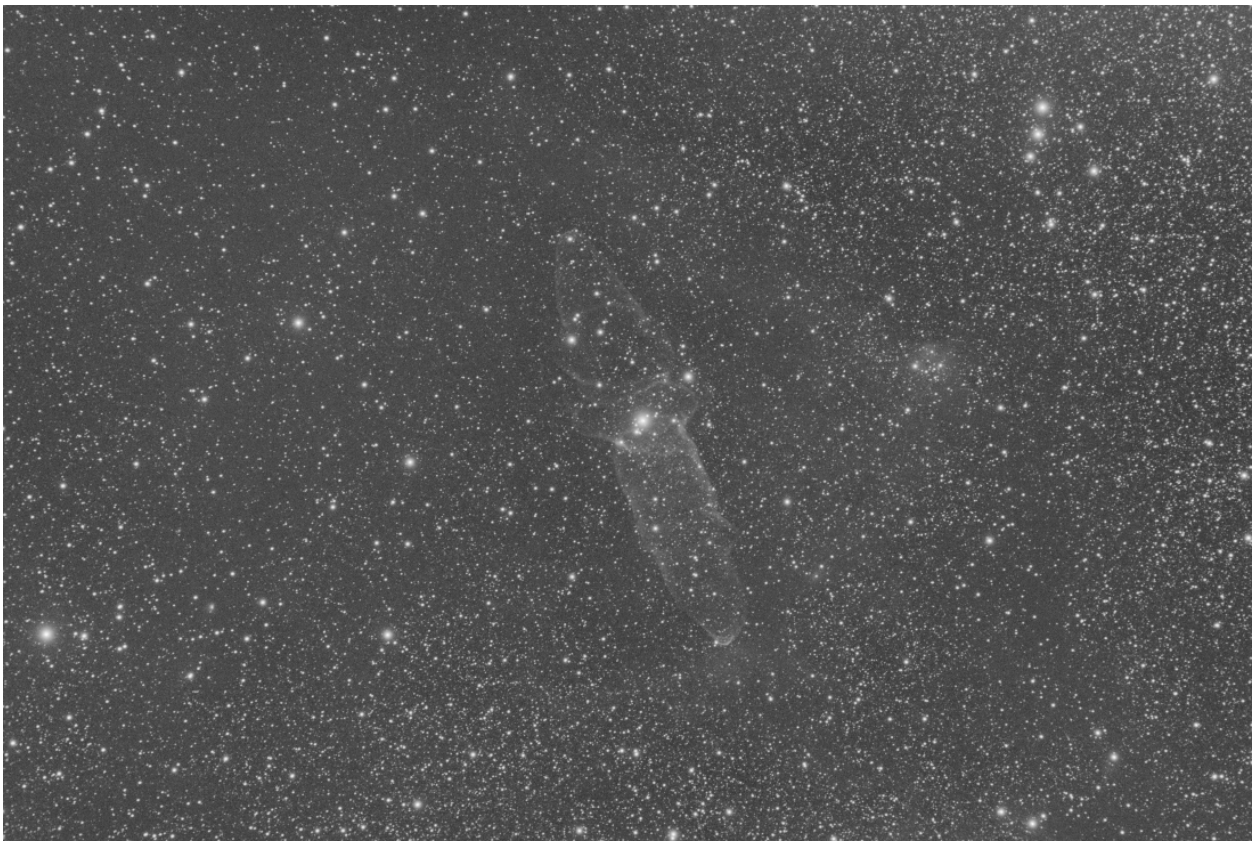


Abb. 26: Originalbild mit schwachem Signal (Bild Cyril Richard).

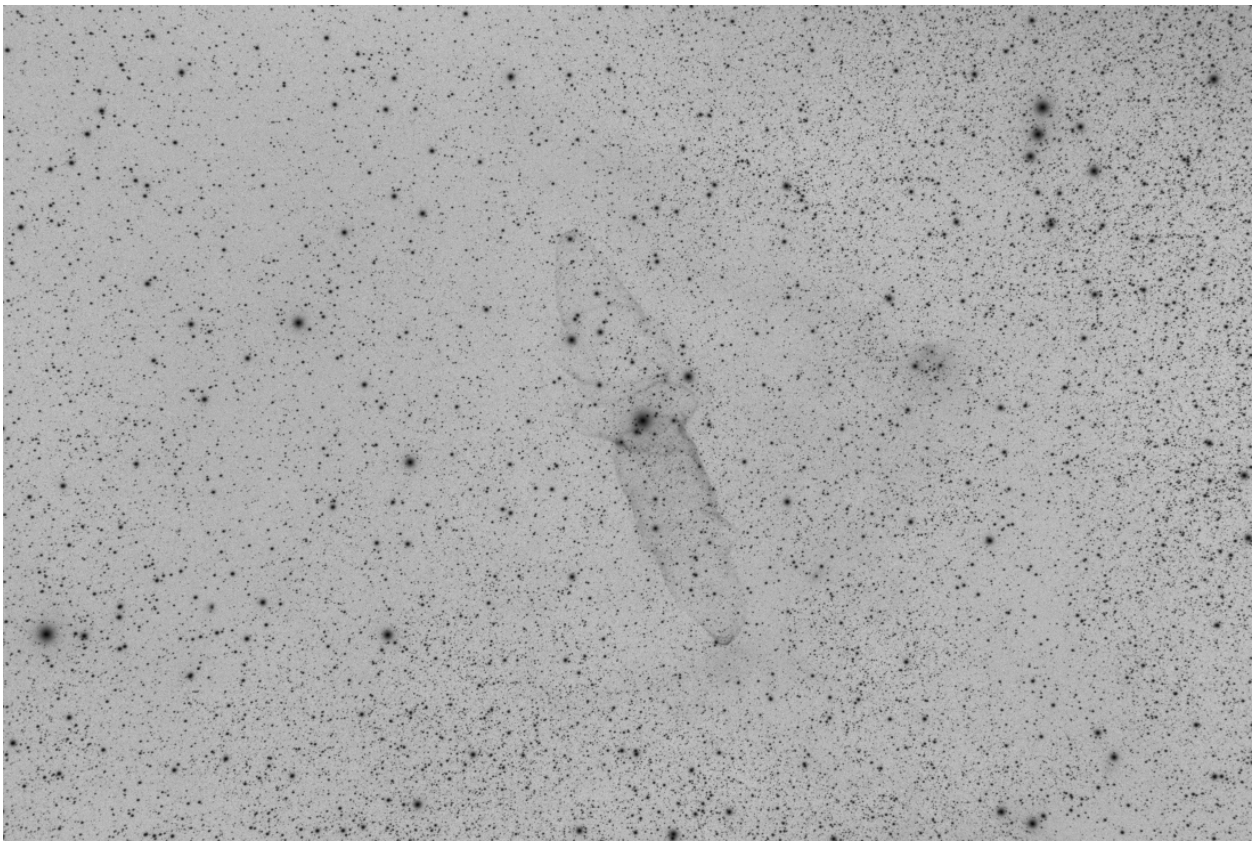


Abb. 27: Negatives Bild, bei dem das Signal besser sichtbar ist (Bild Cyril Richard).

10.4 Filter

In diesem Abschnitt werden alle in Siril vorhandenen Filter vorgestellt. Filter sind Werkzeuge, die die Pixel des Bildes je nach Bedarf verändern.

10.4.1 Wavelet-Transformation "À Trou"

Ein Wavelet ist eine Funktion auf der Grundlage der Wavelet-Zerlegung, einer Zerlegung ähnlich der schnellen Fourier-Transformation, die in der Signalverarbeitung verwendet wird. Es entspricht der intuitiven Vorstellung einer Funktion, die einer kleinen Schwingung entspricht, daher der Name.

Es gibt viele Arten von Wavelet-Funktionen, die ihre eigenen Namen haben, wie in der Abbildung unten dargestellt.

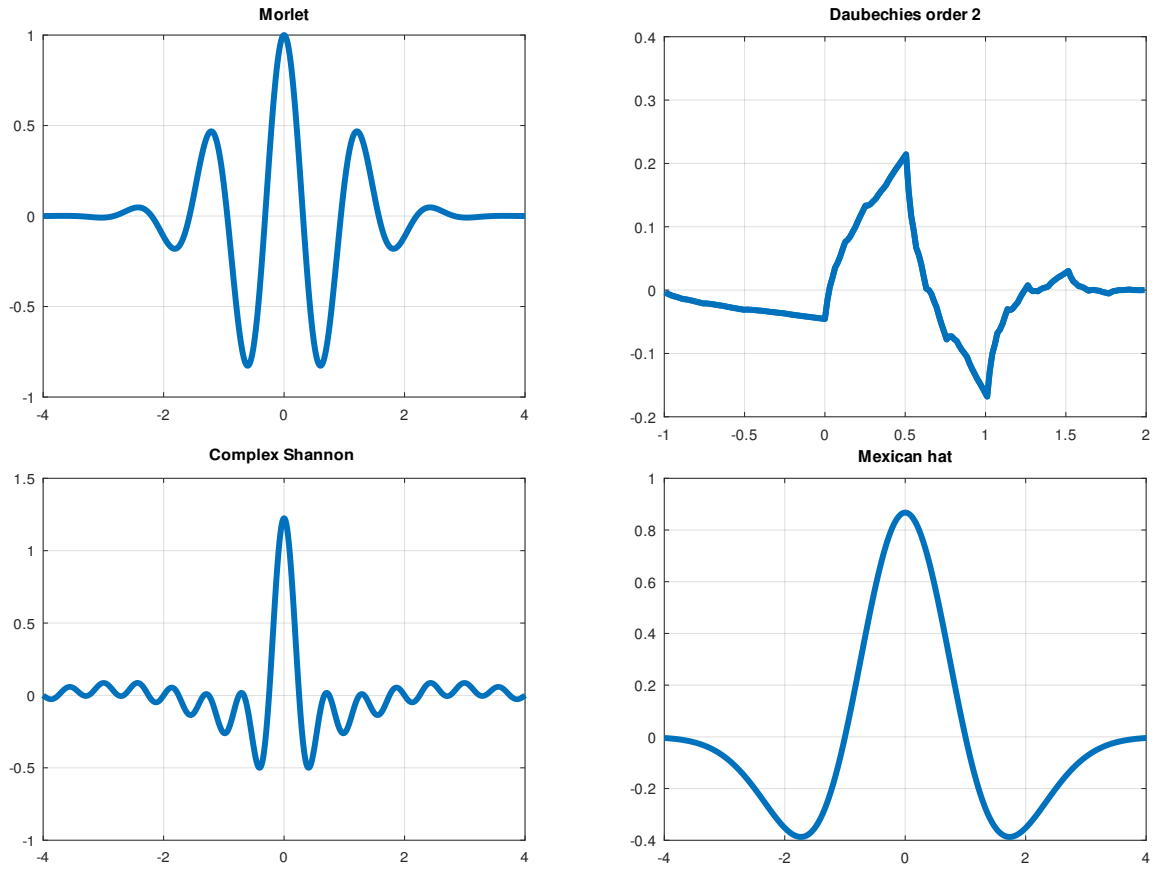


Abb. 28: Ein Beispiel für vier verschiedene Arten von Wavelets.

Die in Siril verwendete À-Trou-Wavelet-Transformation zerlegt ein Bild in eine Reihe von Skalenebenen, die auch als Wavelet-Ebenen bezeichnet werden.

Diese Ebenen können mit dem Extraktionstool *Wavelet Layer* extrahiert werden, allerdings werden sie hier verwendet, ohne visuell zugänglich zu sein. Im Allgemeinen wird dieser Algorithmus häufig am Ende des Stackings von Planetenbildern verwendet. Da das Rauschen ausschließlich in einer der Wavelet-Ebenen enthalten ist, ist es möglich, die Details des Bildes hervorzuheben, indem die Rauschmenge reduziert wird.

Klicken Sie zunächst auf die Schaltfläche *Ausführen*, um die Wavelet-Layer anhand der oben definierten Parameter zu berechnen:

- **Typ**: Es sind zwei Arten von Algorithmen möglich: Linear und BSpline. In der Regel wird der letztere gewählt, auch wenn er etwas langsamer ist.

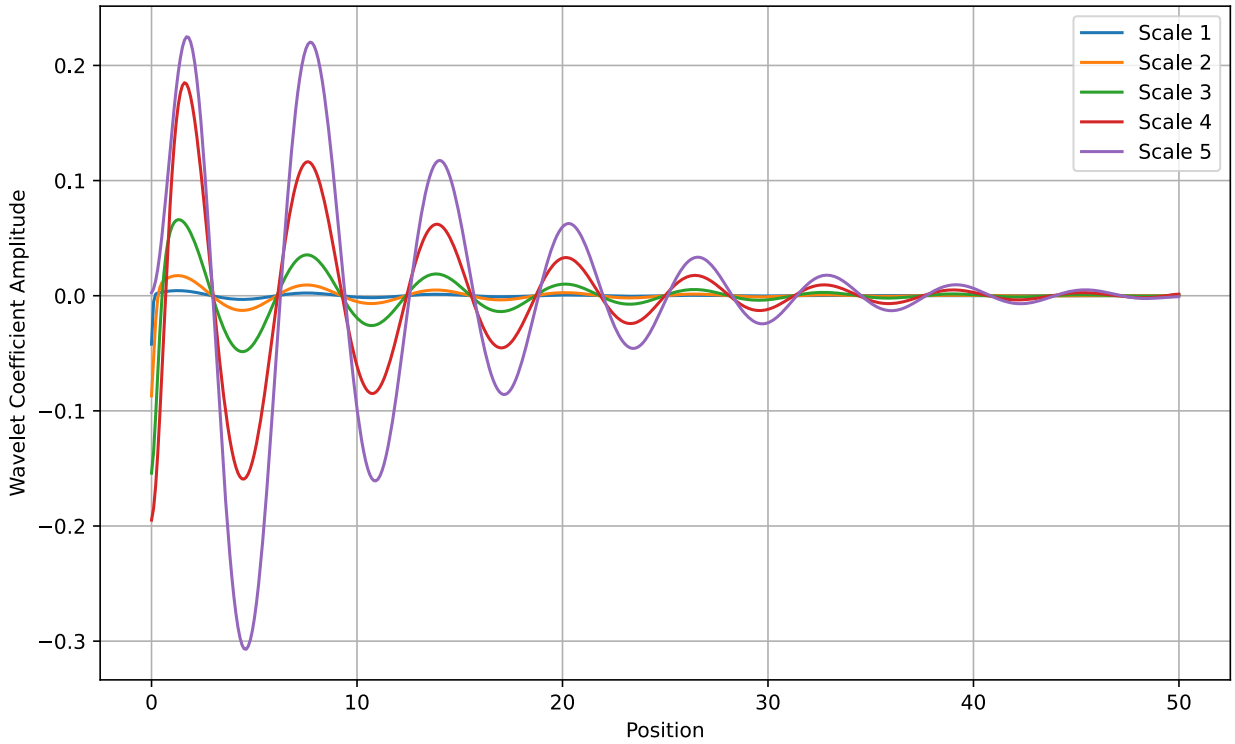


Abb. 29: À Trous Wavelets Transform-Darstellung mit B-Spline-Interpolation.

- **Nb of Layers**: Anzahl der Wavelet-Schichten, die verwendet werden. 6 ist die maximale Anzahl von Ebenen, die verwendet werden kann. Um mit einer größeren Anzahl von Ebenen zu arbeiten, können Sie die unten erläuterte Befehlszeile verwenden.

Jede Ebene hat einen Schieberegler, mit dem der Kontrast dieser Ebene geändert werden kann. Wenn weniger als 6 Ebenen erstellt wurden, sind nur die entsprechenden Schieberegler aktiv. Ein Wert größer als 1 verbessert die Details, während ein kleinerer Wert sie tendenziell verringert.

Dies ist ein Liveview-Tool. Die Änderungen werden in Echtzeit angezeigt und Sie müssen auf *Anwenden* klicken, um sie zu bestätigen. Wenn Sie auf *Zurücksetzen* klicken, werden alle Schieberegler auf 1 zurückgesetzt und somit alle laufenden Änderungen rückgängig gemacht.

Siril Kommandozeile

```
wavelet nbr_layers type
```

Computes the wavelet transform of the loaded image on (**nbr_layers**=1...n) layer(s) using linear (**type**=1) or bspline (**type**=2) version of the 'à trous' algorithm. The result is stored in a file as a structure containing the layers, ready for weighted reconstruction with WRECONS.

Siehe auch EXTRACT

Verweise: *wrecons*, *extract*

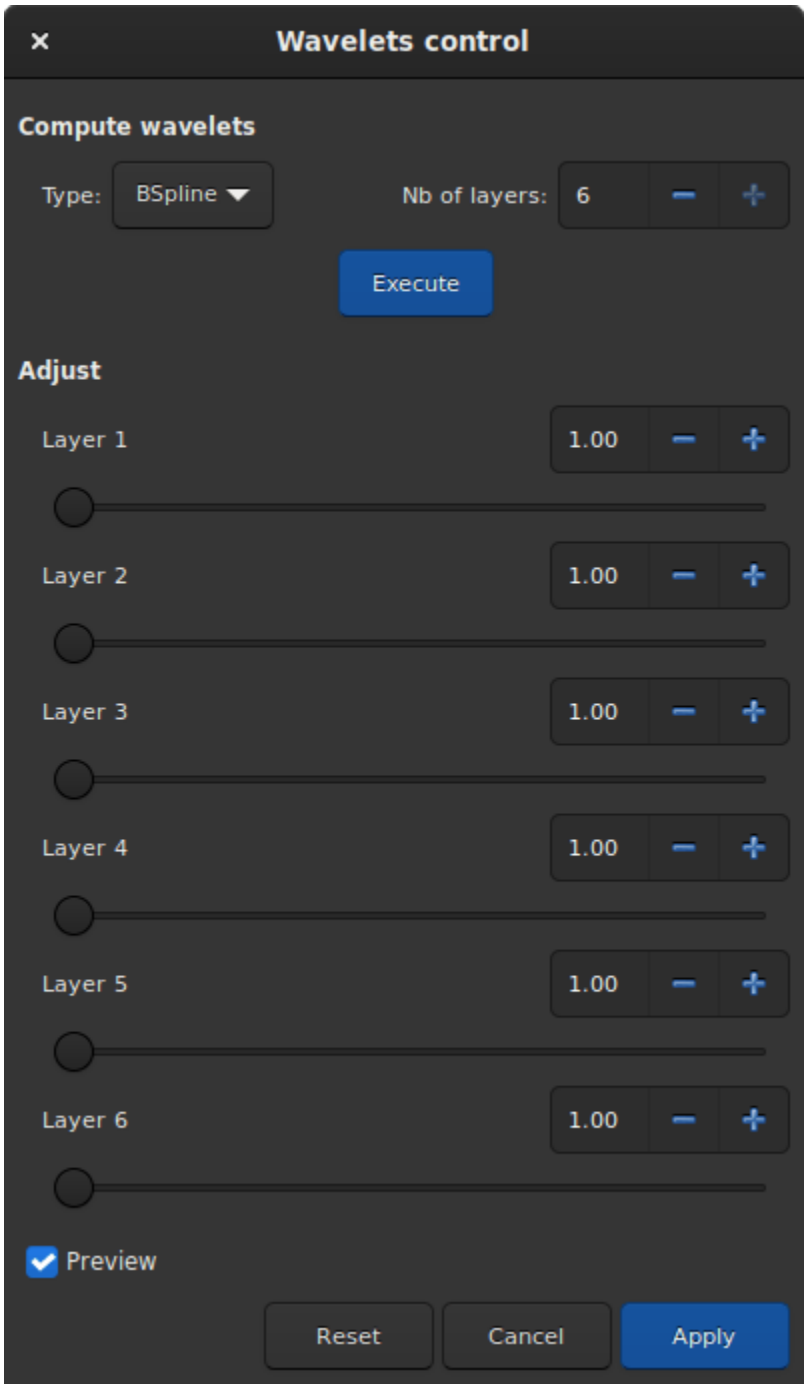


Abb. 30: Dialogfeld "Wavelet-Werkzeug".

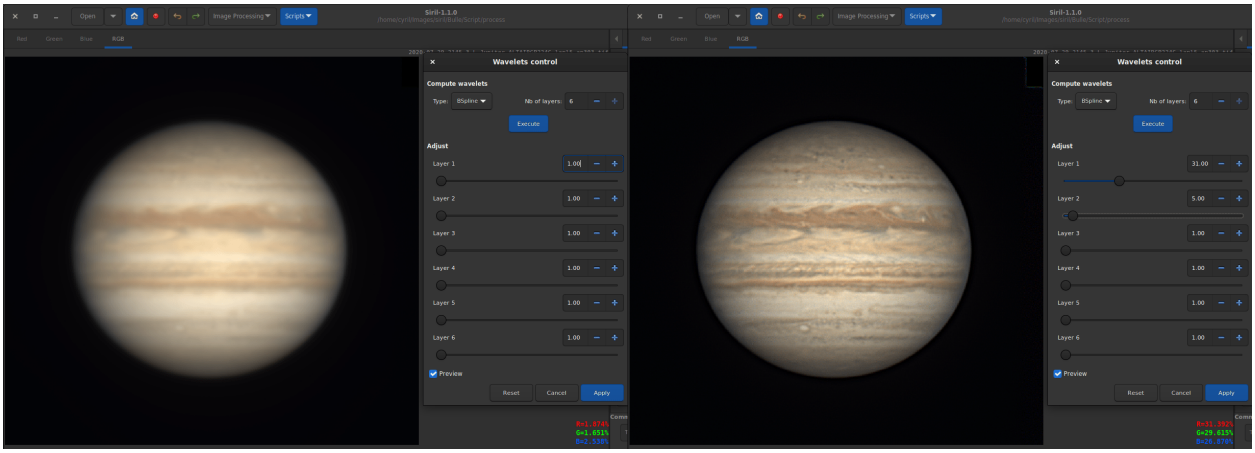


Abb. 31: Anwendung von Wavelets auf ein Jupiter-Bild (mit freundlicher Genehmigung von J.-L. Dauvergne). Das Bild auf der linken Seite ist das Rohbild der Stacking-Ausgabe, während das Bild auf der rechten Seite das gleiche Bild ist, auf das Wavelets angewendet wurden.

Siril Kommandozeile

```
wrecons c1 c2 c3 ...
```

Rekonstruiert das aktuelle Bild aus den Layern, die zuvor mit waveelets berechnet und mit den Koeffizienten **c1**, **c2**, ..., **cn** entsprechend der Anzahl der für die Wavelet-Transformation verwendeten Layer gewichtet wurden, nach der Nutzung von WAVELET

Verweise: *wavelet*

Das Beispiel in der Abbildung oben würde in der Befehlszeile wie folgt geschrieben werden:

```
wavelet 6 2
wrecons 31 5 1 1 1 1
```

10.4.2 Banding Reduzierung

In some cases, images may suffer from a banding defect. This is usually caused by the sensor, or appears in short exposures due to electronic noise; calibration by darks, bias and flats do not improve the images.

Das Fenster zur Reduzierung der Streifenbildung enthält einige Parameter zur Optimierung des Verarbeitungsprozesses:

- **Amount** defines the strength of the correction. The higher the value the stronger the correction, but a value of 1 should be good in most cases as it applies the same correction as ideally computed.
- **Protect from Highlights** will ignore bright pixels when the option is checked, this can be useful for images with large bright objects.
- **Sigma Factor** will adjust the highlight protection. Higher value will ignore only pixels that are brighter.
- **Vertical banding** removes vertical banding instead of the default horizontal banding. This should rarely be needed, but some cameras also have this problem.

Die Anwendung des folgenden Filters auf das Originalbild mit den in der Abbildung gezeigten Parameterwerten führt zu einem schönen Ergebnis ohne Streifenbildung.

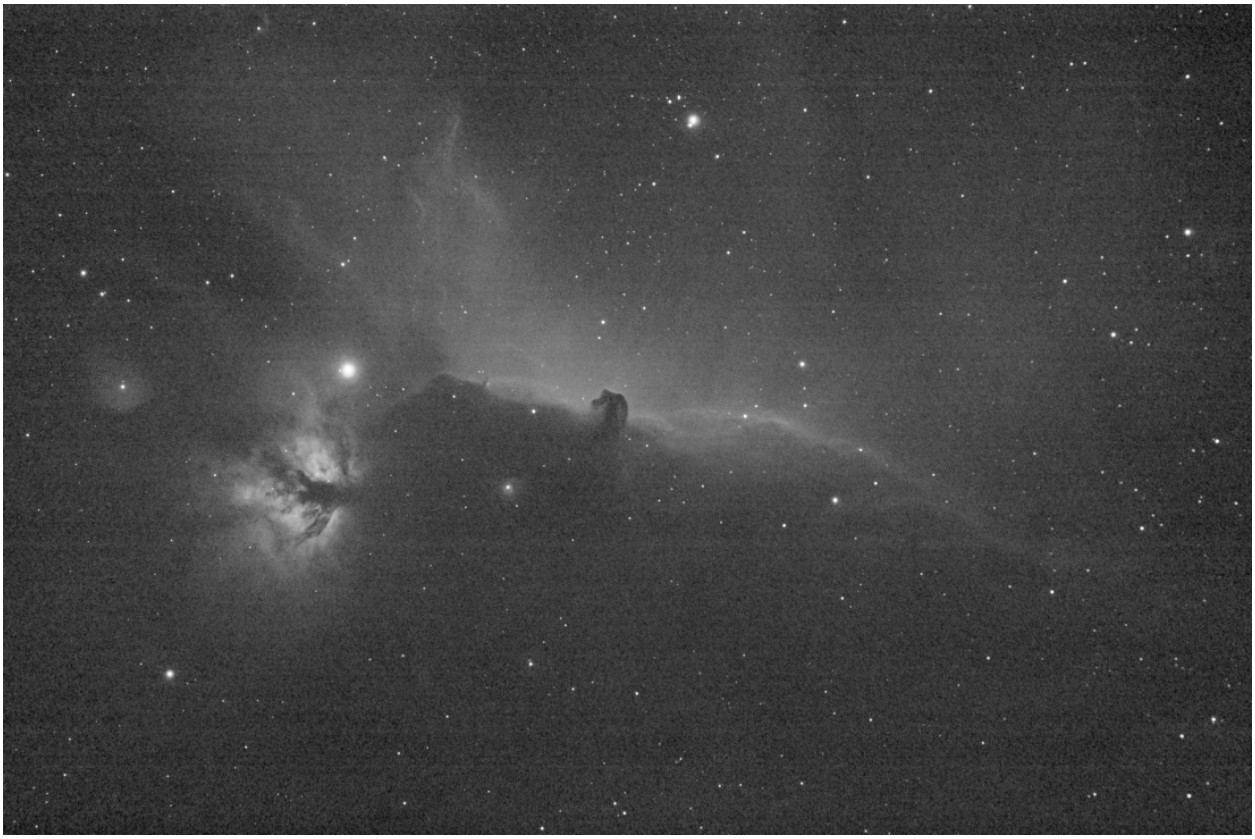


Abb. 32: Original image with visible horizontal banding.

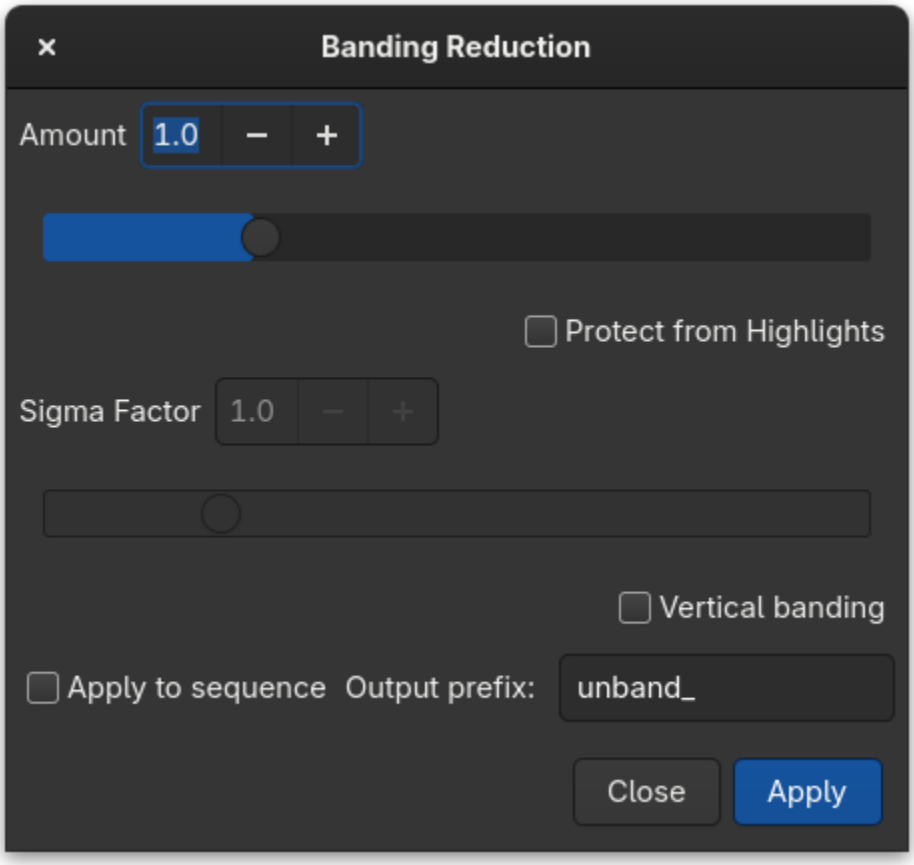


Abb. 33: Dialogbox Banding-Reduzierung.

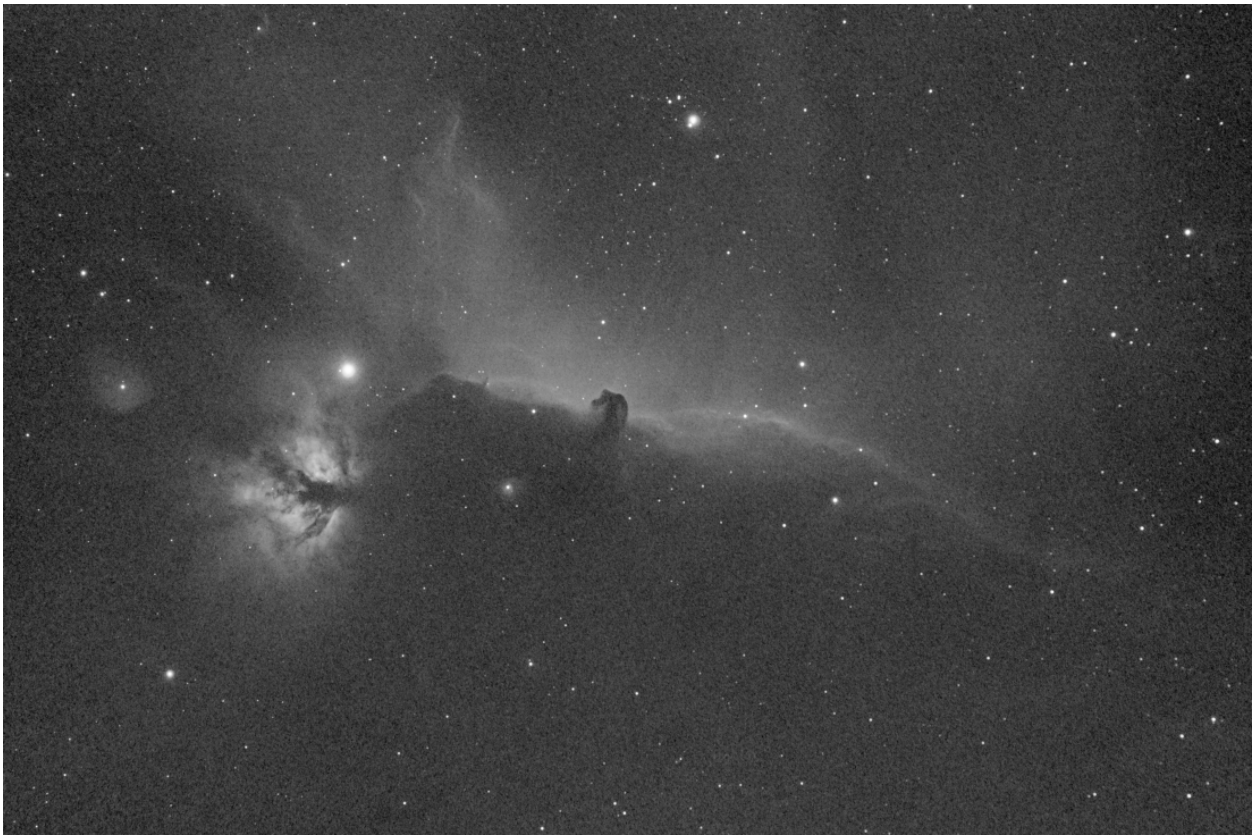


Abb. 34: Ergebnis nach der Durchführung des Filters. Es ist kein Banding mehr sichtbar.

This filter can also be applied to a sequence. Adjust the filter for the loaded image (with a sequence already loaded), then check the *Apply to sequence* button and define the output prefix of the new sequence (`unband_` by default).

Siril Kommandozeile

```
fixbanding amount sigma [-vertical]
```

Versucht das horizontale oder vertikale Banding im geladenen Bild zu entfernen.

Mit dem Argument **amount** wird die Höhe der Korrektur zwischen 0 und 4 festgelegt.

sigma definiert das Hightlight-Schutzniveau des Algorithmus, wobei ein höheres Sigma einen höheren Schutz bietet, der zwischen 0 und 5 liegt. Werte von 1 und 1 sind oft gut genug.

Die Option **-vertical** ermöglicht das Entfernen von vertikalem Banding, die Standardeinstellung ist horizontal

Siril Kommandozeile

```
seqfixbanding sequencename amount sigma [-prefix=] [-vertical]
```

Gleicher Befehl wie FIXBANDING, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "unband_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweis: *fixbanding*

10.4.3 Kontrastbegrenztte Adaptive Histogrammausgleichung (CLAHE)

Die CLAHE-Methode (Contrast-Limited Adaptive Histogram Equalization) wird zur Verbesserung des Kontrasts von Bildern eingesetzt. Sie unterscheidet sich von der gewöhnlichen Histogramm-Streckung dadurch, dass die adaptive Methode mehrere Histogramme berechnet, von denen jedes einem separaten Abschnitt des Bildes entspricht, und diese zur Neuverteilung der Helligkeitswerte des Bildes verwendet. Auf diese Weise kann der lokale Kontrast verbessert und die Kantenschärfe in jedem Bereich eines Bildes erhöht werden.

Tipp

Dieser Filter ist ein Liveview-Filter. Das heißt, dass jede Änderung der Einstellungen automatisch auf dem Bildschirm sichtbar ist. Dies kann jedoch durch Deaktivieren der Schaltfläche *Vorschau* deaktiviert werden.

- Die Größe der Kacheln, in denen die Histogramme berechnet werden, kann über einen Schieberegler festgelegt werden. Standardmäßig ist sie auf 8 eingestellt.
- Die *Obere Grenze* ist eine Option, die verhindert, dass das Rauschen in relativ homogenen Regionen eines Bildes übermäßig verstärkt wird. Dann wird der abgeschnittene Teil des Histogramms, der die Obergrenze überschreitet, gleichmäßig auf alle Intervalle des Histogramms verteilt.

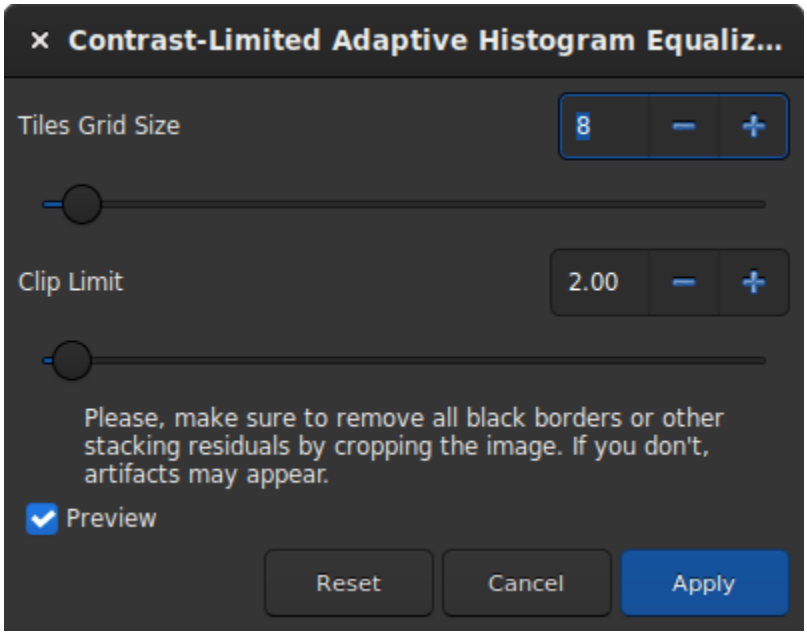


Abb. 35: Dialogbox Kontrastbegrenzte adaptive Histogrammausgleichung.

Tipp

Dieser Filter funktioniert besser bei nicht linearen Daten. Es wird empfohlen, das Bild vorher zu strecken.

Siril Kommandozeile

```
clahe cliplimit tileSize
```

Gleicht das Histogramm eines Bildes mit Hilfe der Kontrastbegrenzten Adaptiven Histogrammangleichung (CLAHE) aus.

cliplimit legt den Schwellenwert für die Kontrastbegrenzung fest.
tilesize legt die Größe des Rasters für die Histogramm-Angleichung fest. Das Eingabebild wird in rechteckige Kacheln gleicher Größe unterteilt

10.4.4 Kosmetische Korrektur

In Siril ist die kosmetische Korrektur der Schritt, der Hot- und Coldpixel im Bild beseitigt. Sie wird in der Regel während der Vorverarbeitung unter Verwendung des Masterdarks durchgeführt. Der Grund dafür ist, dass das Masterdark in der Regel eine gute Karte der defekten Pixel enthält und es einfacher ist, sie darauf zu finden. Wenn Sie jedoch nicht über ein Masterdark verfügen, bietet Siril eine Alternative mit einem automatischen Erkennungsalgorithmus für diese Pixel in einem Light.

Das Dialogfenster enthält mehrere Parameter, die für das ordnungsgemäße Funktionieren des Tools erforderlich sind. Die Verwendung der Standardeinstellungen führt jedoch in der Regel zu guten Ergebnissen.

- **"Kaltes" Sigma**: Wie oft (in durchschnittlichen Abweichungseinheiten) muss ein Pixelwert vom Wert der um-

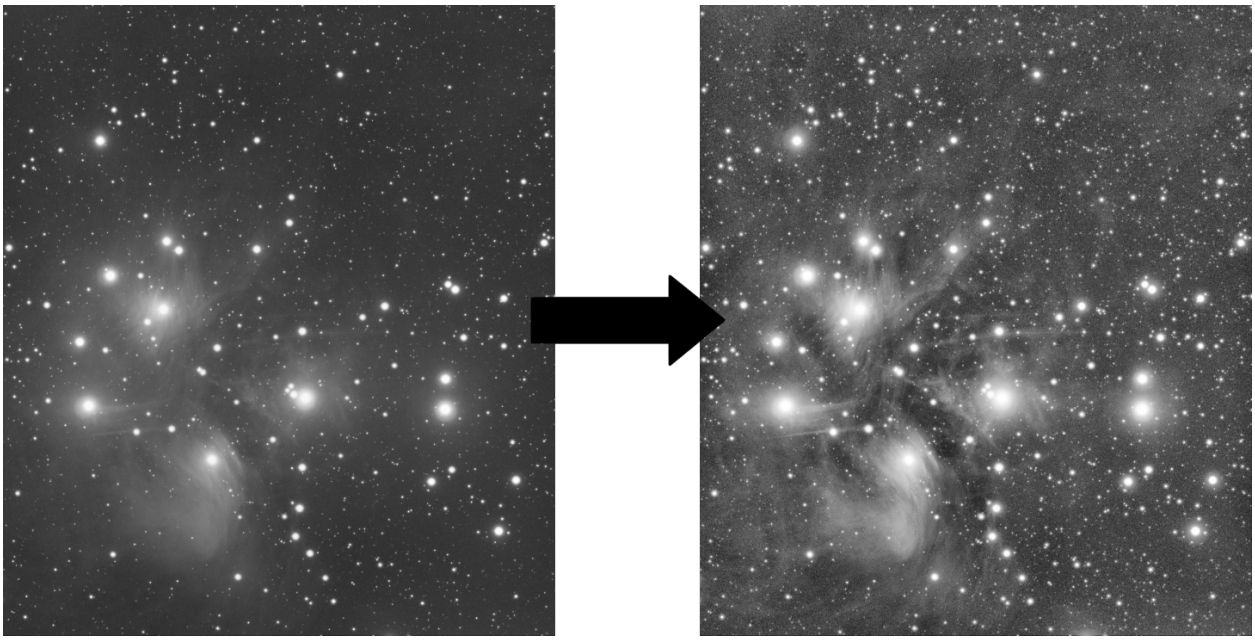


Abb. 36: Ein Beispiel für einen CLAHE-Filter, angewandt auf nicht-lineare Daten mit `Tiles Grid Size=21` und `Clip Limit=4.20`.

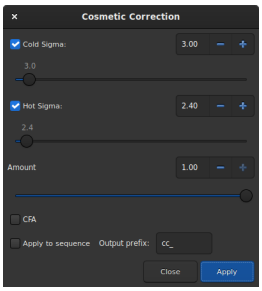


Abb. 37: Dialogfenster Kosmetische Korrektur.

liegenden Nachbarn abweichen, um als Coldpixel zu gelten.

- **"Warmes" Sigma**: Wie oft (in durchschnittlichen Abweichungseinheiten) muss ein Pixelwert vom Wert der umliegenden Nachbarn abweichen, um als Hotpixel zu gelten.
- **Stärke**: Dies ist ein Modulationsparameter, wobei 0 keine Korrektur und 1 eine 100%ige Korrektur bedeutet.
- **CFA**: Diese Option muss für CFA-Bilder mit Bayer-Muster aktiviert werden. Sie funktioniert nicht für den X-Trans-Sensor.

Dieser Vorgang kann auf Sequenzen angewendet werden. Öffnen Sie eine Sequenz und bereiten Sie die Einstellungen vor, die Sie verwenden möchten. Aktivieren Sie dann die Schaltfläche *Auf Sequenz anwenden* und definieren Sie das Ausgabepräfix der neuen Sequenz (standardmäßig `cc_`).

Theorie

Hotpixelerkennung

Nennen wir $m_{5\times5}$ den Median der 5 nächsten Nachbarn. Wenn der Pixelwert größer ist als

$$m_{5\times5} + \max(\text{avgDev}, \sigma_{\text{high}} \times \text{avgDev}),$$

mit avgDev, der *Durchschnittsabweichung* des gesamten Bildes.

Dann wird das Pixel durch den Durchschnitt der $3mal3$ Pixel ersetzt: a_{3mal3}, aber nur wenn

$$a_{3\times3} < m_{5\times5} + \frac{\text{avgDev}}{2}.$$

Coldpixelerkennung

Wenn der Pixelwert kleiner ist als

$$m_{5\times5} - (\sigma_{\text{low}} \times \text{avgDev}),$$

dann wird das Pixel durch $m_{5\times5}$ ersetzt.

Abb. 38: Animation zur kosmetischen Korrektur.

Siril Kommandozeile

```
find_cosme cold_sigma hot_sigma
```

Führt eine automatische Erkennung heißer/kalter Pixel anhand der angegebenen Schwellwerte (ins Sigma-Einheiten) durch

Siril Kommandozeile

```
find_cosme_cfa cold_sigma hot_sigma
```

Gleicher Befehl wie FIND_COSME, aber für CFA-Bilder

Verweis: *find_cosme*

10.4.5 Dekonvolution

Die Dekonvolution oder Entfaltung ist ein mathematisches Werkzeug, um Unschärfe- oder Verzerrungseffekte in einem Bild zu kompensieren. Die wahre Szene ist nicht das, was auf Ihrem Sensor aufgezeichnet wird - Sie nehmen eine Schätzung der wahren Szene auf, die von einer PSF (Point Spread Function) gefaltet wird (mathematisch ausgedrückt, die "unscharfe PSF", die atmosphärische Verzerrungen, physikalische Eigenschaften Ihres Teleskops, Bewegungsunschärfe usw. repräsentiert und Ihre Aufnahme verschlechtert). Die Entfaltung kann diese Bildverschlechterung bis zu einem gewissen Grad rückgängig machen. Es ist jedoch wichtig, von vornherein zu sagen, dass die Entfaltung ein Problem ist, das Mathematiker als *inkorrekt gestellt* oder *schlecht gestellt* (ill-posed) bezeichnen (wie die meisten inversen Probleme). Schlecht gestellt bedeutet, dass es entweder keine Lösung gibt, oder wenn doch, dann ist sie nicht eindeutig, und sie ist nicht kontinuierlich von den Daten abhängig. Im Wesentlichen bedeutet dies, dass die Dekonvolution selbst theoretisch sehr schwierig ist und es keine Garantie dafür gibt, dass sie funktioniert.

All dies wird noch schwieriger, wenn wir nicht genau wissen, was die PSF ist, die wir zu entfernen versuchen. In der Astronomie können wir uns theoretisch ein Bild von der PSF machen, indem wir die Wirkung der Unschärfe auf die Punktquellen (Sterne), die wir abbilden, betrachten. Manchmal ist die tatsächliche PSF jedoch nicht über das gesamte Bild hinweg konstant, manchmal verhindern andere Faktoren wie die Sättigung der Sterne, dass die Stern-PSF eine ganz genaue Schätzung der PSF ist, und manchmal (z. B. bei Mondaufnahmen) gibt es keine Sterne.

Siril zielt darauf ab, einen flexiblen Ansatz zur Dekonvolution zu bieten. Es gibt mehrere Optionen für die Definition oder Schätzung der PSF und mehrere Entfaltungsalgorithmen, die für die letzte Phase der Entfaltung nach der Definition der PSF zur Auswahl stehen.

Abb. 39: Beispiel einer Dekonvolution eines Sternfeldes.

Der Zugriff auf die Entfaltung erfolgt über das Menü *Bildverarbeitung* oder über Siril-Befehle.

Überblick über die Nutzung

- Um eine Dekonvolution-PSF zu erzeugen, wählen Sie die gewünschte PSF-Erzeugungsmethode aus und drücken Sie *Erzeuge die PSF*. Dies kann getrennt von der eigentlichen Dekonvolution durchgeführt werden, so dass der Benutzer die Auswirkungen der Änderung der PSF-Parameter sehen kann.
- Siril erzeugt nur monochrome PSFs, da dies der häufigste Anwendungsfall ist und die Benutzeroberfläche vereinfacht. Es können jedoch drei monochrome PSFs gespeichert und zusammengesetzt werden, um eine 3-Kanal-PSF zu erzeugen, die geladen und zur Dekonvolution von 3-Kanal-Bildern verwendet werden kann.
- Um die Dekonvolution auf ein einzelnes Bild anzuwenden, wählen Sie die gewünschte PSF-Erzeugungsmethode aus und drücken Sie *Anwenden*. Wenn zuvor eine Methode zur blinden PSF-Schätzung ausgeführt wurde, wird die Methode automatisch auf *Vorherige PSF* gesetzt, um eine unnötige Neuberechnung der PSF zu vermeiden.
- Um die Entfaltung auf eine Sequenz anzuwenden, gehen Sie wie oben beschrieben vor, stellen aber sicher, dass Sie das Kontrollkästchen *Auf Sequenz anwenden* aktivieren. Sie können auch ein benutzerdefiniertes Präfix für die Ausgabesequenz angeben: Wenn kein anderes Präfix angegeben wird, wird das Standardpräfix (`dec_`) verwendet.
- Beim der Dekonvolution einer Sequenz wird die PSF nur für das erste Bild berechnet. Dieselbe PSF wird für alle Bilder der Sequenz wiederverwendet.

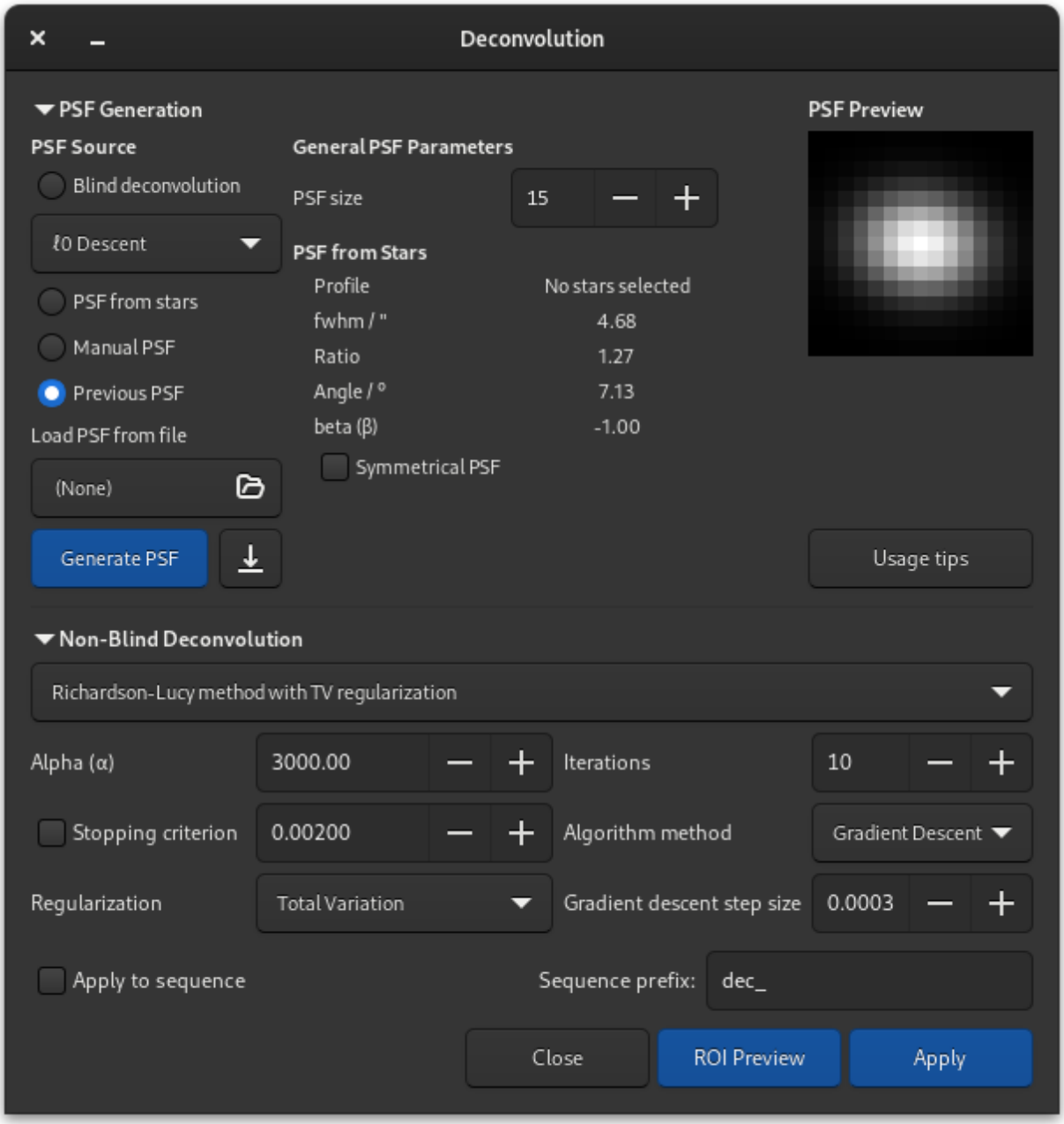


Abb. 40: Dialogbox des Dekonvolution-Werkzeugs.

Überblick über die Methoden zur Definition des Unschärfekerns

- $_0$ **Abstieg**: Dies ist die Standardmethode zur PSF-Schätzung, die auf der Arbeit von Anger, Delbracio und Facciolo basiert. Die Parameter müssen im Allgemeinen nicht angepasst werden, außer dass Sie bei besonders großen PSFs vielleicht das Modell zur mehrstufigen PSF-Schätzung ausprobieren möchten. Mehrstufig ist standardmäßig deaktiviert, da während der Entwicklung festgestellt wurde, dass es bei den häufigeren kleinen bis mittleren PSF-Größen zu eher unnatürlichen Ergebnissen führt.
- **Spektrale Unregelmäßigkeit** [Goldstein2012]: Diese Methode zur PSF-Schätzung wird als Alternative angeboten. Im Allgemeinen schneidet sie nicht so gut ab wie die $_0$ Gradientenabstiegs-Methode, sie kann jedoch nützlich sein, wenn Sie ein Bild entdecken, bei dem die Standardmethode keine guten Ergebnisse liefert. Für diese Methode muss das latent scharfe Bild keine Kanten enthalten, solange das spektrale Zerfallsmodell eingehalten wird. Der Gradientenabstieg $_0$ hingegen geht von einem ähnlichen Modell aus (da die Kanten den gleichen spektralen Zerfall aufweisen), setzt aber voraus, dass die Gradienten spärlich und kontrastreich sind, d. h. dass die Kanten gleichphasig sind, so dass dieses Modell theoretisch bei kontrastarmen, sternlosen Bildern besser funktionieren könnte. Es sind wahrscheinlich einige Experimente erforderlich, um den Algorithmus zu finden, der am besten zu Ihren Daten passt.
- **PSF aus den Sternen**: Diese Methode modelliert eine PSF aus der durchschnittlichen PSF der ausgewählten Sterne. Es ist wichtig, bei der Auswahl der Sterne wählerisch zu sein: Sie dürfen nicht gesättigt sein, da dies zu einer groben Verzerrung der PSF-Schätzung führen würde, aber sie dürfen auch nicht so schwach sein, dass die Sternanalysefunktionen von Siril ungenaue Messungen der Sterne liefern. Die ausgewählten Sterne sollten einigermaßen hell sein, sich ziemlich zentral im Bild befinden und in einem Bereich des Bildes mit einem ziemlich konstanten Hintergrund liegen. Sobald die Sterne ausgewählt sind, können Sie entweder ein Gauß- oder ein Moffat-Sternprofilmodell wählen. Bei der Ausführung der Dekonvolution wird die PSF aus den durchschnittlichen Parametern der ausgewählten Sterne synthetisiert. Wenn keine Sterne ausgewählt werden, versucht Siril, Sterne mit einer Spitzenamplitude zwischen 0,07 und 0,7 mit einem Moffat-Profil automatisch zu erkennen. Dieser Bereich vermeidet gesättigte Sterne sowie solche, die zu schwach sind, um eine genaue Lösung zu liefern, und liefert im Allgemeinen gute Ergebnisse.
- **Manuelles PSF**: Mit dieser Methode können Sie eine PSF manuell definieren. Es können Gauß-, Moffat-, Scheiben- oder Beugungsscheibchen- PSF-Modelle definiert werden. Beachten Sie, dass die FWHM in Pixeln und nicht in Bogensekunden angegeben wird. Die Gauß- und Moffat-Modelle eignen sich für die Entfaltung der Formen von Sternen, die durch atmosphärische Verzerrungen entstehen; die Scheiben-PSF-Modelle eignen sich für die Dekonvolution des Effekts einer leichten Unschärfe.
- **Lade PSF aus einer Datei**: Mit dieser Methode können Sie ein PSF aus einem beliebigen von Siril unterstützten Bildformat laden. Die bereitgestellte PSF muss quadratisch sein (wenn sie nicht quadratisch ist, wird sie abgelehnt) und sollte ungerade sein (wenn sie nicht ungerade ist, wird sie um ein Pixel in jeder Richtung beschnitten, was jedoch eine leicht dezentrierte PSF ergibt und im Vergleich zur Bereitstellung einer ungeraden PSF nicht optimal ist). Es können entweder monochrome oder 3-Kanal-PSFs geladen werden. Wenn eine 3-Kanal-PSF in Verbindung mit einem monochromen Bild geladen wird, werden die gleich gewichteten Luminanzwerte der PSF verwendet. Wenn eine 3-Kanal-PSF zusammen mit einem 3-Kanal-Bild geladen wird, wird jeder Kanal des Bildes unter Verwendung des entsprechenden Kanals der PSF dekonvolviert. Wenn eine monochrome PSF zusammen mit einem 3-Kanal-Bild geladen wird, wird das Bild in den LAB-Farbraum konvertiert und der L-Kanal (Luminanz) wird unter Verwendung der monochromen PSF dekonvolviert, um die Berechnungen effizienter zu gestalten.
- **Vorherige PSF**: Diese Methode ermöglicht die Wiederverwendung der zuvor geschätzten Unschärfe-PSF. Sie ist vor allem bei den Methoden zur blinden PSF-Schätzung von Nutzen: Wenn Sie mit der geschätzten PSF zufrieden sind, aber eine Reihe von Testläufen mit verschiedenen Parametern für die letzte Stufe der Dekonvolution durchführen möchten, können Sie die vorherige PSF wiederverwenden und etwas Rechenzeit sparen.
- Nach der Schätzung können die PSFs auf Wunsch gespeichert werden. Wenn Siril mit libtiff-Unterstützung kompiliert wurde, wird der PSF im 32-Bit-TIFF-Format gespeichert, mit dem gleichen Dateinamen wie das aktuelle Bild, aber mit einem Datums- und Zeitstempel versehen und mit dem Suffix `_PSF`. Wenn Siril ohne libtiff-Unterstützung gebaut wurde, wird das PSF als FITS-Datei gespeichert. Während dies das primäre Format von

Siril für astronomische Bilddateien ist, wird TIFF für PSFs bevorzugt: Der Nachteil der Verwendung des FITS-Formats für PSFs ist die möglicherweise eingeschränkte Kompatibilität mit Bildbearbeitungsprogrammen, die Sie zur Bearbeitung oder Untersuchung der gespeicherten Datei verwenden möchten.

Tipp

Während die blinde Dekonvolution einer Entfaltungs-PSF bei linearen und nichtlinearen Daten möglich ist, kann die Verwendung einer PSF aus Stern-PSF nur bei linearen Bildern erfolgen. Andernfalls wären die PSF-Werte nicht gültig.

Tipp

If a ROI is set, blind PSF estimation methods will calculate the PSF from the ROI rather than the whole image. If you do wish to calculate a PSF from the whole image you must clear the ROI before estimating the PSF, then set the ROI to apply a preview. However, you also have the option of calculating the PSF only from a selected part of the image. This may be desirable if you have optical aberrations close to the edge of the image and wish to estimate the PSF from the central area only.

Überblick über die nicht-blinde Dekonvolution

- **Richardson-Lucy-Dekonvolution** [Lucy1974]: Dies ist der standardmäßige, nicht-blinde Dekonvolutionsalgorithmus. Es handelt sich um ein iteratives Verfahren, das für seine Verwendung bei der Korrektur von Bildverzerrungen in der Anfängen der Betriebszeit des Hubble-Weltraumteleskops bekannt ist. In Siril wird es entweder durch die Methode der totalen Variation (TV) reguliert, die darauf abzielt, den Algorithmus für die Verstärkung des Rauschens zu bestrafen, oder durch die Frobenius-Norm der lokalen Hessian-Matrix. Diese Regularisierung basiert auf den zweiten Ableitungen. Neben der Regularisierung gibt es ein Stoppkriterium, mit dem der Algorithmus frühzeitig gestoppt werden kann, sobald seine Konvergenzrate unter ein bestimmtes Niveau fällt. Eine Erhöhung des Wertes des Stoppkriteriums kann die Ringbildung (Ringing) um Sterne und kontraststarke Kanten reduzieren. Es werden zwei Formulierungen des Richardson-Lucy-Algorithmus angeboten: die multiplikative Formulierung und die Formulierung des Gradientenabstiegs. Letztere ermöglicht eine bessere Kontrolle, da die Schrittgröße des Gradientenabstiegs geändert werden kann (der Nachteil dabei ist, dass durch die Verwendung kleinerer Schritte mehr Iterationen erforderlich sind, um das gleiche Konvergenzniveau zu erreichen). Der größte Vorteil der Gradientenabstiegsmethode ist, dass sie eine stärkere Regularisierung ermöglicht - dies kann beim multiplikativen Richardson-Lucy-Algorithmus problematisch sein, da der Regularisierungsterm im Nenner erscheint und kleine Werte hier (starke Regularisierung) zu Instabilität führen können. Siril verwendet die naive Faltung für kleine Kernelgrößen und die FFT-basierte Faltung für größere Kernelgrößen, bei denen FFTs einen effizienteren Algorithmus bieten. (Dies geschieht automatisch und erfordert keinen Benutzereingriff.)
- **Wiener Filter-Methode**: Diese Methode ist eine nicht-iterative Entfaltungsmethode. Sie modelliert ein angenommenes Gaußsches Rauschprofil, d.h. Rauschen, das durch ein konstantes Profil modelliert wird. Die Konstante Alpha wird verwendet, um die Stärke der Regularisierung in Bezug auf den Rauschpegel festzulegen. Wie bei den anderen Algorithmen sorgt ein kleinerer Wert von Alpha für eine stärkere Regularisierung. Dieser Algorithmus eignet sich gut für Mondbilder, bei denen das Rauschen Gauß- und nicht Poisson-Charakter hat, funktioniert aber in der Regel schlecht bei Deep Space-Bildern, bei denen das Rauschen meist dazu tendiert einer Poisson-Verteilung zu entsprechen.
- **Split-Bregman-Methode**: Diese Methode wird intern im Rahmen der Unschärfe-PSF-Schätzung verwendet und wird auch als Entfaltungsalgorithmus für die letzte Stufe angeboten. Es ist ein häufig verwendeter Algorithmus für die Lösung konvexer Optimierungsprobleme. Auch dieser Algorithmus wird durch eine Kostenfunktion für die Gesamtvariation reguliert. Er schneidet bei Bildern mit Sternen nicht so gut ab wie Richardson-Lucy, kann aber für sternlose Bilder oder Bilder von der Mondoberfläche in Betracht gezogen werden.

Tipp

Die Wahl der Entfaltungsmethode ist sehr wichtig, um gute Ergebnisse zu erzielen. Bei DSO-Bildern ist es im Allgemeinen richtig, eine Richardson-Lucy-Methode zu verwenden: Sowohl die Split-Bregman- als auch die Wiener-Methode liefern aufgrund des extremen Dynamikbereichs um Sterne herum schlechte Ergebnisse. Bei linearen Bildern ist es in der Regel am besten, die Richardson-Lucy-Methode mit Gradientenabstieg zu verwenden, und wenn in der Nähe heller Sterne Ringing auftritt, sollte die Schrittweite verringert werden. Dieser Ansatz verringert die Auswirkung jeder Iteration, so dass mehr Iterationen erforderlich sind, aber es bedeutet, dass Sie eine feinere Kontrolle erreichen können, indem Sie die Dekonvolution nur bis zu dem Punkt durchführen, an dem sich Artefakte bilden, und dann ganz leicht zurückgehen. Für gestreckte Bilder können Sie den multiplikativen Richardson-Lucy-Algorithmus verwenden.

Tipp

Für gestackte Mond- und Planetenbilder können die Split-Bregman- oder Wiener-Methoden besser geeignet sein. Diese Methoden erfordern im Allgemeinen keine Iteration wie die Richardson-Lucy-Methode und sind möglicherweise besser für die Rauscheigenschaften von gestackten Bildern mit hohem Signal-Rausch-Verhältnis geeignet. (Der Richardson-Lucy-Algorithmus basiert auf der Annahme eines Poisson-Rauschens, was in der Regel auf DeepSky-Bilder zutrifft, während die hier implementierte Wiener-Methode von einer Gaußschen Rauschverteilung ausgeht, die besser zu gestackten Planeten-/Mondbildern passt.)

Parameter und Einstellungen

Allgemeine Einstellungen

- Größe der PSF. Die PSF-Eingangsgröße sollte ausreichend groß gewählt werden, um sicherzustellen, dass die PSF im angegebenen Bereich enthalten ist. Wird sie jedoch zu groß gewählt, können die Methoden zur blinden PSF-Schätzung zu einem schlechteren und zeitaufwändigeren Ergebnis führen.
- Lambda (λ). Regularisierungsparameter für die PSF-Schätzung. Versuchen Sie, diesen Wert für verrauschte Bilder zu verringern.

Einstellung zur PSF Schätzung des $_0$ Gradientenabstiegs

- Mehrstufig. Diese Einstellung aktiviert die mehrstufige PSF-Schätzung. Dies kann helfen, die PSF-Schätzung zu stabilisieren, wenn eine große PSF-Größe angegeben wird, aber einige PSFs, die mit dieser Option erzeugt werden, können zu unnatürlich aussehenden Ergebnissen führen, weshalb sie standardmäßig deaktiviert ist.
- Experteneinstellungen. Diese sollten normalerweise nicht angepasst werden müssen, werden aber für Neugierige zur Verfügung gestellt.
  - Gamma legt die Stärke der Regularisierung fest, die bei der Durchführung der Vorhersage des geschärften Bildes verwendet wird. Bei einem gegebenen Gamma wird mit zunehmendem Rauschen auch die Schätzung stärker verrauscht. Wenn Gamma erhöht wird, wird die Schätzung weniger durch Rauschen beeinträchtigt, ist aber tendenziell glatter. Der Standardwert von 20 wurde in [Anger2019] experimentell ermittelt.
  - Iterationen legt die Anzahl der Iterationen fest, die bei der PSF-Schätzung verwendet werden. Die Autoren des Algorithmus berichten, dass es nur einen minimalen Vorteil bringt, diese Zahl auf 3 zu erhöhen, und dass es überhaupt keinen Vorteil bringt, sie über 3 hinaus zu erhöhen.
  - Lambda-Verhältnis und Lambda-Minimum legen die Parameter für die Verfeinerung der Schärfevorhersage durch aufeinanderfolgende Werte des Regularisierungsparameters für die Vorhersage des geschärften Bildes bei jeder Iteration der Methode fest.

  - Skalierungsfaktor, Hochskalierungsunschärfe und Herunterskalierungsunschärfe werden nur verwendet, wenn die mehrstufige Schätzung aktiv ist. Sie legen den Standard-Skalierungsfaktor zwischen den einzelnen Skalierungsstufen und den Grad der Unschärfe fest, der bei der Neuskalierung zwischen den einzelnen Skalen verwendet wird.
  - Kernel-Schwellenwert. Werte unterhalb dieser Schwelle werden bei der PSF-Schätzung auf Null gesetzt.

Spektrale Unregelmäßigkeit Einstellungen für die PSF-Schätzung

- Der Kompensationsfaktor steuert die Stärke eines Filters, der verwendet wird, um eine übermäßige Schärfe in der geschätzten PSF zu vermeiden. Für Bilder mit intrinsischer Unschärfe sollte ein Wert nahe bei Eins verwendet werden. Bei an sich scharfen Bildern können niedrige Werte zu Artefakten führen und der Wert sollte auf eine große Zahl erhöht werden, um den Filter effektiv zu deaktivieren.
- Experteneinstellungen. Diese sollten normalerweise nicht angepasst werden müssen, werden aber für Neugierige zur Verfügung gestellt.
  - Iterationen der inneren Schleife legt die Anzahl der Iterationen fest, die in der inneren Schleife der spektralen Unregelmäßigkeitsmethode durchgeführt werden. Der Algorithmus konvergiert schnell und es kann möglich sein, die Anzahl der Iterationen auf etwa 100 zu reduzieren, ohne dass sich das Ergebnis wesentlich verschlechtert.
  - Abtastungen der äußerer Schleife. Hier wird festgelegt, wie viele Zufallsphasen abgetastet werden sollen. Da die Phasenabfrage mit zufälligen Werten für jede Probe beginnt, ist es wichtig, genügend Proben zu ziehen, um zu vermeiden, dass ein lokales Minimum erreicht wird. Die PSF stabilisiert sich bei Bildern mit geringem Rauschen schnell, aber wenn Sie mit dieser Methode bessere Ergebnisse erzielen möchten, sollten Sie dies als erste der Experteneinstellungen ausprobieren, insbesondere bei Bildern mit höherem Rauschen.
  - Iterationen der äußeren Schleife. [Anger2018], legt nahe, dass 2 Iterationen ausreichen können, um eine plausible PSF-Schätzung zu erhalten, und dass es unnötig ist, diesen Wert über 3 zu erhöhen.

PSF aus den Sternen

- Diese Methode der PSF-Erzeugung hat keine einstellbaren Parameter. Sie erzeugt eine PSF basierend auf den durchschnittlichen Parametern der ausgewählten Sterne mit dem Befehl *findstar* oder dem Dialog *Dynamische PSF*. Die durchschnittlichen Parameter werden im Dekonvolutionsdialog angezeigt, wenn diese PSF-Erzeugungsmethode gewählt wird. Es ist besser, wenn der Benutzer die Sterne, die er für diese Methode verwenden möchte, aktiv auswählt, um eine möglichst genaue PSF zu erhalten. Idealerweise sollten etwa 10 ziemlich helle, aber nicht gesättigte Sterne aus dem zentralen Bereich des Bildes ausgewählt werden (um Sterne auszuschließen, die möglicherweise unter Koma oder anderen Aberrationen leiden). Wenn der Benutzer jedoch keine Sterne ausgewählt hat, versucht Siril, geeignete Sterne automatisch zu erkennen, indem es seine Erkennungsroutine mit Filtern durchführt, die so eingestellt sind, dass nur Sterne mit Spitzenamplituden zwischen 0,07 und 0,7 berücksichtigt werden. Dieser Bereich vermeidet sowohl gesättigte Sterne als auch solche, die zu schwach sind, um eine genaue Lösung zu liefern. Dies funktioniert in den meisten Fällen gut, kann aber dennoch durch außermittige Aberrationen beeinträchtigt werden.
- Wenn Sie das Kontrollkästchen *Symmetrische PSF* aktivieren, wird die erzeugte PSF perfekt kreisförmig sein. Dies entspricht dem durchschnittlichen FWHM und Beta der ausgewählten Sterne, aber nicht der Elongation.

Manuelle PSF

Diese Methode der PSF-Erzeugung ermöglicht die Erzeugung einer benutzerdefinierten parametrischen PSF.

- Der Profiltyp ermöglicht die Wahl des PSF-Profils. Es werden Gauß-, Moffat-, Scheiben- und Beugungsscheibchen-PSFs unterstützt.

- Gauß- und Moffat-PSFs werden für den Abgleich der aus dem Bild gemessenen Sternparameter verwendet. Sie sollten eine gute Schätzung der gesamten Unschärfefunktion liefern, die auf das Bild angewendet wird, da Sterne Punktlichtquellen sind.

Abb. 41: Ein Beispiel für Moffat-PSF mit fwhm=5", Winkel=45°, Verhältnis=1,20, $\beta = 4,5$ und einer PSF-Größe von 15.

- Scheiben-PSFs werden zur Dekonvolution von Bildern verwendet, die unscharf sind.

Abb. 42: Ein Beispiel für ein Scheibenprofil mit fwhm=5" und einer PSF-Größe von 15.

- PSFs von Beugungsscheibchen werden verwendet, um die Beugung, die als physikalische Folge der Lichtbeugung durch die Öffnung Ihres Teleskops auftritt, zu korrigieren.

Abb. 43: Beispiel einer Beugungsscheibchen-PSF mit Teleskopdurchmesser=250mm, Brennweite=4500mm, Wellenlänge=525nm, Pixelgröße=2,9µm, zentraler Obstruktion=40% und einer PSF-Größe von 41.

- FWHM gibt die volle Breite bei halbem Maximum des gewählten Profils an (bei Scheiben-PSFs wird einfach der Radius angegeben).
- Beta (β) gibt den im Moffat-PSF-Profil verwendeten Beta-Parameter an. Er wird bei anderen PSF-Profilen ignoriert.
- Für Beugungsscheibchen-PSFs sind eine Reihe von Parametern Ihres Teleskops und Sensors erforderlich:
  - Öffnung
  - Brennweite
  - Sensor-Pixelgröße
  - Zentrale Wellenlänge, die abgebildet wird. Siril versucht, diese Daten aus den Metadaten Ihres Bildes zu extrahieren, wenn sie verfügbar sind. Wenn jedoch einige Parameter fehlen oder fehlerhaft erscheinen, hebt Siril sie hervor und gibt eine Warnung im Protokoll aus, die Ihnen empfiehlt, sie zu überprüfen. Das Verhältnis der zentralen Obstruktion ist ebenfalls erforderlich, um ein genaues Beugungsscheibchen zu erzeugen. Dieser wird als Prozentsatz ausgedrückt, d. h. die Gesamtfläche der zentralen Obstruktion geteilt durch die Gesamtfläche der Öffnung x 100. Bei Refraktoren ist dieser Wert gleich Null, bei anderen Teleskopen variiert er: Er kann bei einem Newton-Reflektor etwa 20 % und bei einigen korrigierten Dall-Kirkham-Teleskopen sogar 40-50 % betragen. Sie müssen Ihr Instrument ausmessen oder die Spezifikationen des Herstellers konsultieren.

Richardson-Lucy Dekonvolution

Die Parameter zur Konfiguration der Richardson-Lucy-Entfaltung in Siril lauten wie folgt:

- **alpha** legt die Stärke der Regularisierung fest. Ein kleinerer Wert von alpha führt zu einer stärkeren Regularisierung und einem glatteren Ergebnis; ein größerer Wert verringert die Stärke der Regularisierung und bewahrt mehr Bilddetails, kann aber zu einer Verstärkung des Rauschens führen.

- **Iterationen** gibt die maximale Anzahl der zu verwendenden Iterationen an. Wenn kein Rauschen vorhanden ist, führt eine große Anzahl von Iterationen dazu, dass die Entfaltung die Schätzung näher an das wahre Bild heranführt. Eine zu große Anzahl von Iterationen verstärkt jedoch auch das Rauschen und verursacht Ringing-Artefakte um Sterne. Die Voreinstellung ist 1 Iteration: eine höhere Zahl kann eingestellt werden, um mehrere Iterationen automatisch zu berechnen, oder Sie können *Anwenden* drücken, um eine Iteration nach der anderen anzuwenden, bis Sie mit dem Ergebnis zufrieden sind. (Oder gehen Sie noch einen Schritt weiter, entscheiden Sie, dass Sie nicht mehr zufrieden sind und verwenden Sie *Undo*.)
- **Stoppkriterium** legt ein Konvergenzkriterium auf der Grundlage aufeinander folgender Schätzungsdifferenzen fest. Damit wird der Algorithmus angehalten, sobald die Konvergenz innerhalb der angegebenen Grenze liegt. Dies ist ein wichtiger Parameter - wenn Sie Ringe um Sterne in Ihrem endgültigen Bild erhalten, versuchen Sie, den Wert des Stoppkriteriums zu erhöhen. Sie können diese Funktion auch ganz deaktivieren, indem Sie das Kontrollkästchen abwählen.
- **Algorithmus/Mmethode** gibt an, ob die multiplikative Implementierung oder die Implementierung des Gradientenabstiegs verwendet werden soll.
- **Schrittgröße des Gradientenabstiegs** gibt die Schrittweite für die Implementierung des Gradientenabstiegs an. Sie sollte nicht zu groß gewählt werden, da der Algorithmus sonst nicht konvergieren kann. Dieser Parameter hat keine Auswirkung, wenn die multiplikative Implementierung ausgewählt ist.

> **Tipp**
>
> Bei linearen Bildern bieten die Methoden des Gradientenabstiegs die notwendige Kontrolle, um Ringing um Sterne herum zu vermeiden. Bei der Dekonvolution von gestreckten Bildern kann dies jedoch unnötig langsam sein, so dass die Verwendung der multiplikativen Methoden oft Zeit sparen kann, ohne die Bildqualität zu beeinträchtigen.

Split Bregman Dekonvolution

Die Parameter für die Konfiguration der Split-Bregman-Entfaltung in Siril lauten wie folgt:

- **alpha** legt die Stärke der Regularisierung fest. Ein kleinerer Wert von alpha führt zu einer stärkeren Regularisierung und einem glatteren Ergebnis; ein größerer Wert verringert die Stärke der Regularisierung und bewahrt mehr Bilddetails, kann aber zu einer Verstärkung des Rauschens führen.
- **Iterationen** gibt die maximale Anzahl der zu verwendenden Iterationen an. Die Split-Bregman-Methode erfordert nicht mehrere Iterationen in der hier implementierten Form, kann aber auf Wunsch iteriert werden. Dies macht im Allgemeinen nur einen kleinen Unterschied und ist daher standardmäßig auf 1 eingestellt.

Wiener Filter

Die Wiener Entfaltung in Siril erfordert nur einen Parameter:

- **alpha** legt die Stärke der Regularisierung fest. Ein kleinerer Wert von alpha führt zu einer stärkeren Regularisierung und einem glatteren Ergebnis; ein größerer Wert verringert die Stärke der Regularisierung und bewahrt mehr Bilddetails, kann aber zu einer Verstärkung des Rauschens führen.

FFTW Leistungseinstellungen

Die Algorithmen zur PSF-Schätzung und -Entfaltung machen ausgiebig Gebrauch von schnellen Fourier-Transformationen (FFT) unter Verwendung der FFTW-Bibliothek. Dies bietet eine Reihe von Einstellungsmöglichkeiten, die auf der Registerkarte "Speichereinstellungen" des Siril-Hauptdialogs *Einstellungen* angepasst werden können.

Hinweis zur Reihenfolge der Bildzeilen

Bei verschiedenen Arten von Bildern, die von Siril verarbeitet werden, können die Pixeldaten in unterschiedlicher Reihenfolge angeordnet sein. SER-Videodateien speichern die Daten immer von oben nach unten, während FITS-Dateien die Daten entweder von unten nach oben oder von oben nach unten speichern können. Die ursprüngliche Empfehlung lautet "von unten nach oben", allerdings stammen FITS-Dateien zunehmend von CMOS-Kameras, bei denen die Pixelreihenfolge eher von oben nach unten verläuft.

Wenn ein Bild mit einer PSF dekonvolviert wird, die aus demselben Bild (oder bei geöffnetem Bild) erstellt wurde, ist dies kein Problem. Es kann jedoch zu Problemen kommen, wenn eine PSF mit einem Bild mit einer Reihenfolge erzeugt und zur Dekonvolvierung eines Bildes oder einer Sequenz mit der entgegengesetzten Reihenfolge verwendet wird. Dies ist ein Nischenanwendungsfall, aber die konsequente Handhabung führt zu einem Verhalten, das auf den ersten Blick überraschend sein kann: Es wird daher im Folgenden erläutert.

Siril behandelt das Problem, indem es die Zeilenreihenfolge des Bildes verfolgt, mit dem das PSF erstellt wurde. PSFs werden immer in der Reihenfolge von unten nach oben gespeichert (wobei sie automatisch umgedreht werden, wenn sie mit einem Bild von oben nach unten erstellt wurden), und wenn sie geladen werden, wird die Reihenfolge an die Reihenfolge des aktuell geöffneten Bildes angepasst. Wird ein Bild mit der entgegengesetzten Zeilenreihenfolge geöffnet, wird die Zeilenreihenfolge des PSF entsprechend geändert. Das bedeutet, dass, wenn Sie z. B. einige FITS-Bilder von unten nach oben nehmen, eines davon zur Erzeugung einer PSF verwenden und sie dann in eine SER-Sequenz von oben nach unten konvertieren, die PSF in die richtige Ausrichtung konvertiert wird, um mit der SER-Sequenz übereinzustimmen. Wenn eine PSF in der Vorschau angezeigt wird, während ein Bild mit der umgekehrten Reihenfolge geöffnet wird, wird die Vorschau nicht sofort aktualisiert: Die Änderung der Reihenfolge wird automatisch erkannt und die PSF zu dem Zeitpunkt gespiegelt, zu dem sie auf das Bild angewendet wird.

Verbrecheralbum

In diesem Abschnitt werden einige Beispiele gezeigt, bei denen die Dekonvolution schief gelaufen ist, und es wird erklärt, warum.

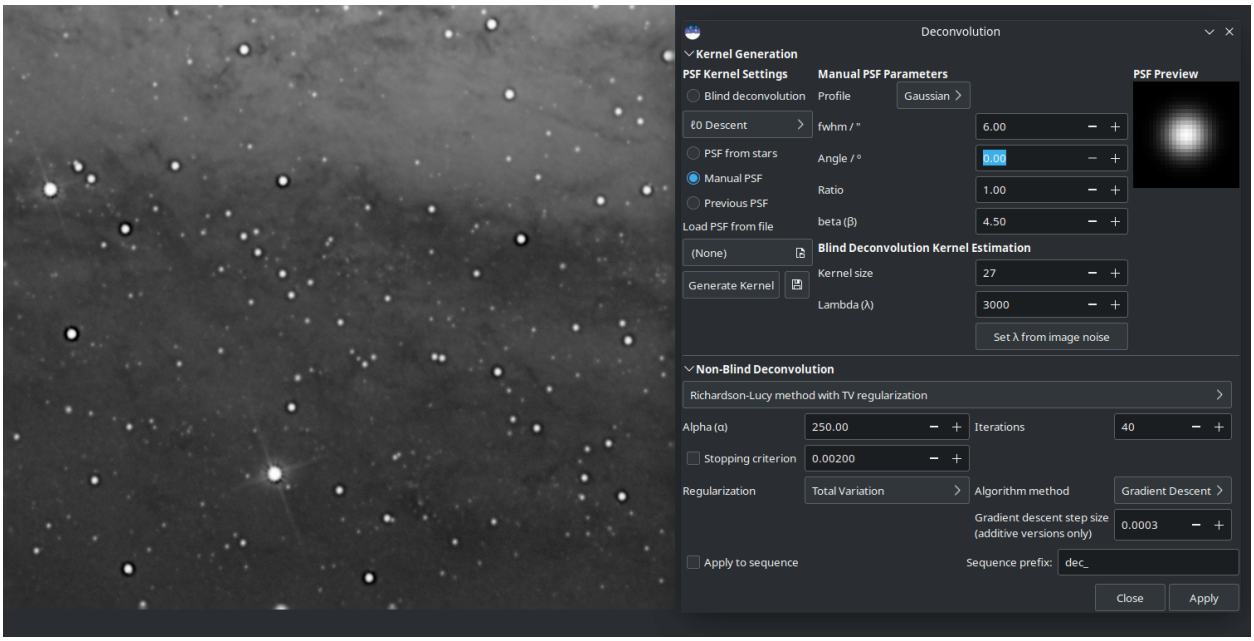


Abb. 44: Die manuell festgelegte PSF war zu groß, was zu großen dunklen Ringen um die Sterne führte.

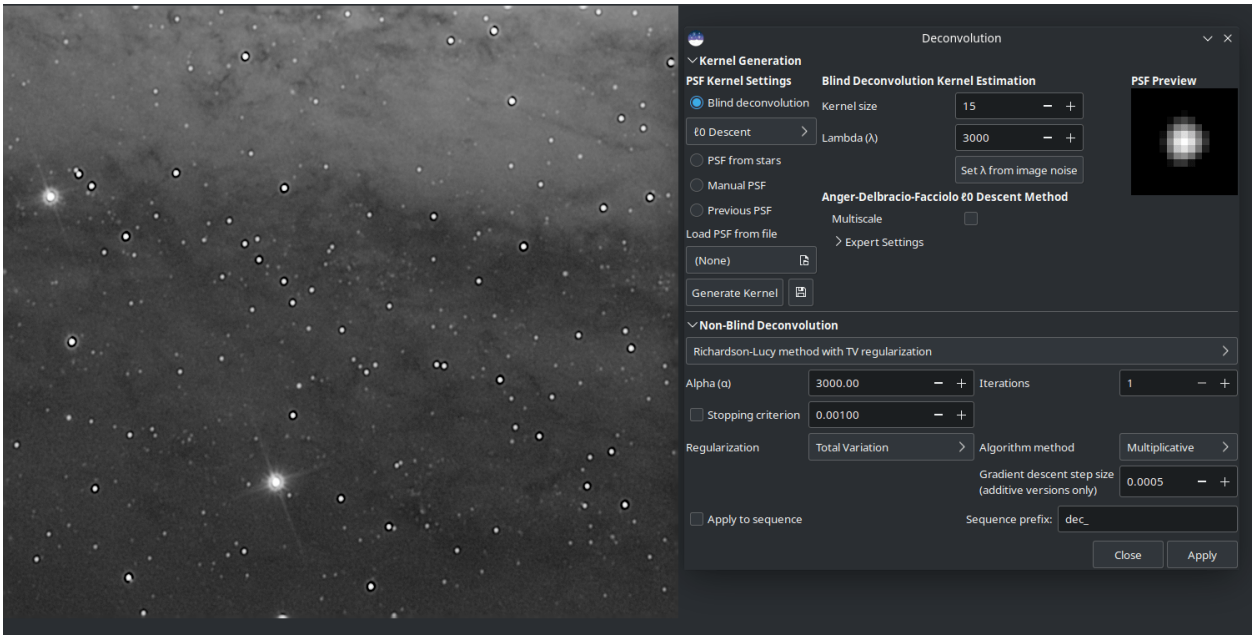


Abb. 45: Es wurden zu viele Iterationen durchgeführt. (Ich habe sie einzeln angewandt, um das Ergebnis zu übertreiben, weshalb der Iterationsparameter immer noch 1 anzeigt.)

Dekonvolution: Tipps für die Verwendung

Sie sind über die Schaltfläche "Hinweise" im Entfaltungswerkzeug in Siril hierher gelangt. Keine Sorge: Dekonvolution ist eine knifflige Technik. Selbst in der mathematischen Theorie ist sie sehr schwierig: Es gibt keine Garantie dafür, dass das Verfahren immer zu einer eindeutigen Lösung konvergiert, die Ihr Bild verbessert. Daher finden Sie hier einige Tipps, wie Sie die Dekonvolution von Siril optimal nutzen können.

Welchen PSF soll ich verwenden?

Die Verwendung einer präzisen PSF ist von grundlegender Bedeutung, um gute Ergebnisse bei der Dekonvolution zu erzielen. Die beiden einfachsten Möglichkeiten zur Erzeugung einer PSF sind die Verwendung einer blinden PSF-Schätzung oder die Modellierung der PSF anhand von Sternen im Bild.

PSF aus den Sternen

Siril kann Sterne in Ihrem Bild erkennen und modellieren. Einzelheiten finden Sie auf der Handbuchseite zu *Dynamische PSF*. Um ein gutes Modell für Ihre PSF zu erhalten, versuchen Sie, das Moffat-Sternprofil in Dynamische PSF auszuwählen. Sterne sind Punktlichtquellen, so dass die Streufunktion eines durchschnittlichen Sterns ein gutes Modell für die Unschärfeeffekte ist, die wir durch Entfaltung zu entfernen versuchen.

Tipp

Sobald Sie Sterne gefunden haben, sortieren Sie sie nach der Spitzenamplitude (Parameter "A"). Wählen Sie alle Sterne mit einer Amplitude von mehr als 0,7 oder weniger als 0,1 aus und löschen Sie sie, und wenn Ihr Bild Hintergrundgalaxien enthält, überprüfen Sie, dass keine falsch-positiven Sterne übrig bleiben. Sterne in diesem Helligkeitsbereich sind nicht gesättigt und nicht zu schwach, um ein genaues PSF-Modell zu liefern.

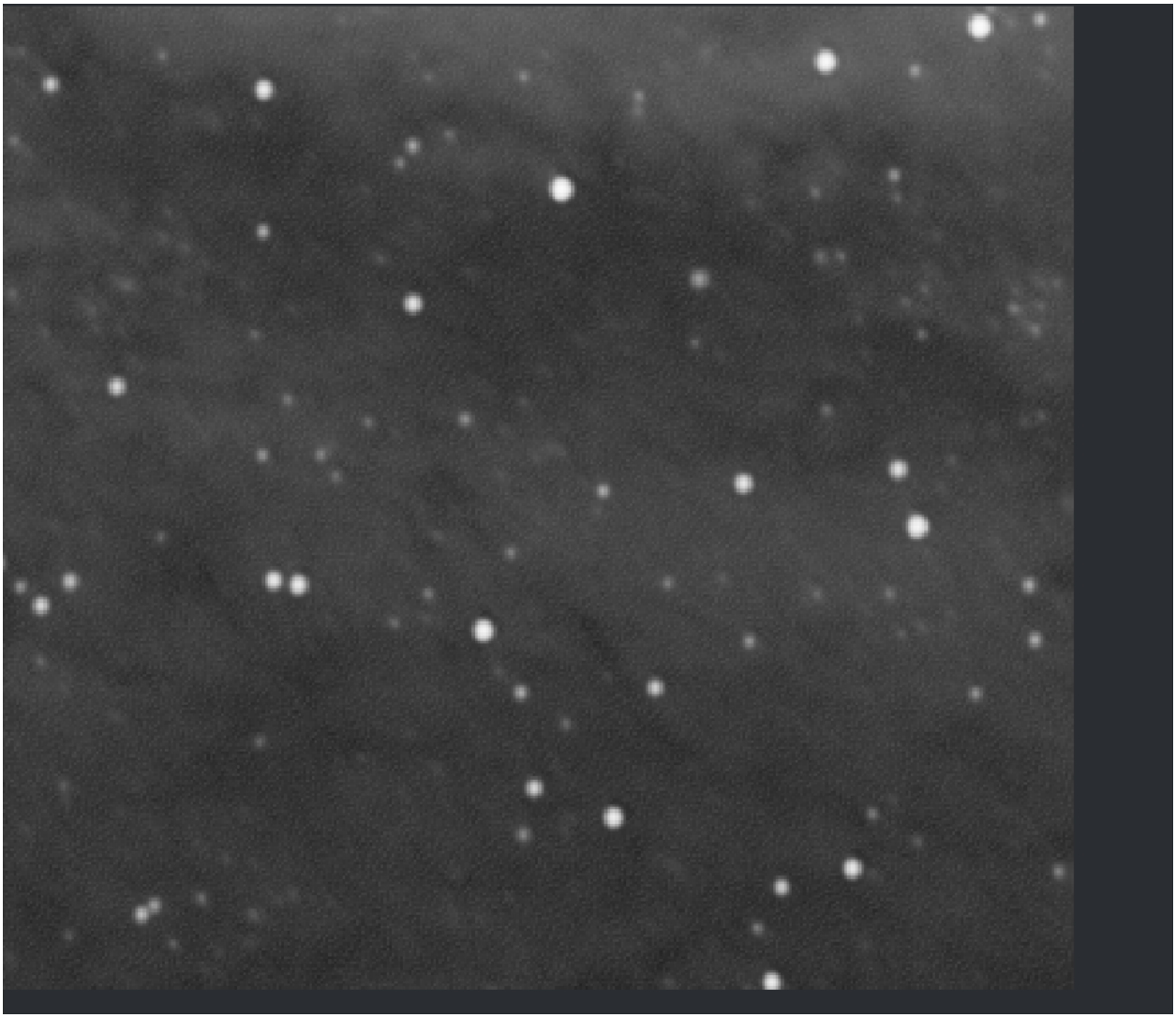


Abb. 46: Eine Nahaufnahme zeigt die Auswirkungen einer zu starken Regularisierung (Alpha = 30) mit der multiplikativen Version von Richardson-Lucy. Für eine starke Regularisierung und / oder eine bessere Kontrolle über jede Iteration wird die Formulierung des Gradientenabstiegs empfohlen.

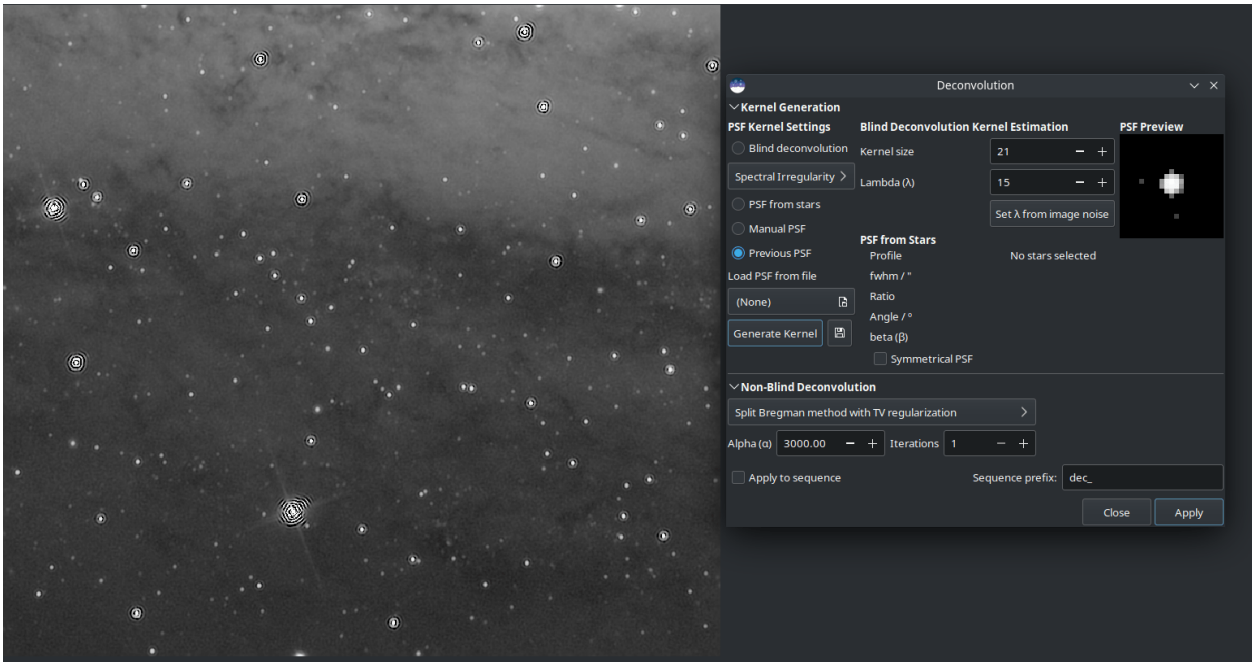


Abb. 47: Typisches Beispiel für den Versuch, ein nicht gestrecktes Sternfeld mit Split Bregman (in diesem Fall) oder Wiener Filter zu dekonvolvieren. Diese sind besser für Planeten-/Mond-/Sonnenbilder geeignet; für Sternenlandschaften wird immer Richardson-Lucy empfohlen.

> **Tipp**
>
> Während die blinde Dekonvolution einer Entfaltungs-PSF bei linearen und nichtlinearen Daten möglich ist, kann die Verwendung einer PSF aus Stern-PSF nur bei linearen Bildern erfolgen. Andernfalls wären die PSF-Werte nicht gültig.

Blinde PSF-Schätzung

Diese Methoden können eine PSF automatisch auf der Grundlage des Bildes selbst schätzen. Wenn Sie keine besseren Vorkenntnisse über die PSF haben, wie z. B. Sterne im Bild (z. B. Mondbilder, die keine Sterne enthalten), ist dies möglicherweise die beste Option. In den meisten Fällen ist es empfehlenswert, die Standardmethode $_0$ zu verwenden: Sie ist schneller und liefert in der Regel bessere Ergebnisse.

> **Tipp**
>
> Wie auch immer Sie Ihre PSF erzeugen, prüfen Sie in der Vorschau, ob sie nicht abgeschnitten aussieht. Wenn dies der Fall ist, erhöhen Sie die PSF-Größe, bis keine wesentlichen Teile des PSF mehr abgeschnitten werden.

Andere PSF-Erzeugungsmethoden

Andere erwähnenswerte Methoden zur PSF-Erzeugung sind das manuelle Scheibenprofil und die Beugungsscheibchen Das Scheibenprofil kann verwendet werden, um Bilder zu verbessern, bei denen der Fokus minimal verschoben ist. Versuchen Sie, die Größe der Scheibe an die Stärke der Unschärfe anzupassen. Das Beugungsscheibchen kann verwendet werden, um die leichte Unschärfe zu korrigieren, die durch die Beugung der Teleskopoptik selbst verursacht wird.

> **Tipp**
>
> Wenn Sie über ein hervorragendes Seeing verfügen (wenig bis keine atmosphärische Unschärfe), reicht es vielleicht aus, das Bild mit einem Beugungsscheibchen zu entfalten.

Dekonvolvierung des Bildes

Sobald Sie eine PSF erzeugt haben, mit der Sie zufrieden sind, können Sie Ihr Bild entfalten. Um gute Ergebnisse zu erzielen, ist es wichtig, die richtigen Einstellungen zu verwenden.

> **Tipp**
>
> Deconvolution is quite slow for large images. To make it quicker to find the best parameters, save your work at this point and use the ROI feature.

Bilder mit Sternen

Bilder, die Sterne enthalten, insbesondere lineare (nicht gestreckte) Daten, sollten immer mit der Richardson-Lucy-Methode entfaltet werden. Ignorieren Sie Split Bregman und Wiener: Diese Algorithmen sind besser für Bilder des Sonnensystems geeignet.

Bei der Dekonvolution von Deep-Sky-Aufnahmen gibt es zwei Schwierigkeiten: Ringing um helle Sterne und Rauschverstärkung im Hintergrund.

Um die Ringe um die Sterne zu vermeiden, verwenden Sie die Methode des Gradientenabstiegs und erhöhen Sie die Anzahl der Iterationen allmählich, bis Sie Anzeichen für die Bildung dunkler Ringe um die Sterne sehen, und reduzieren Sie dann die Iterationen etwas.

Die obige Animation zeigt, wie sich die Verringerung der Anzahl der Iterationen der multiplikativen Formulierung von Richardson-Lucy auswirkt: Sie veranschaulicht auch die feinere Steuerung, die durch die Verwendung der Gradientenabstiegsmethode erreicht werden kann, allerdings auf Kosten von mehr Iterationen.

Um die Verstärkung des Hintergrundrauschens in den Griff zu bekommen, können Sie versuchen, vor der Entfaltung eine kleine Rauschunterdrückung anzuwenden. Wählen Sie im Dialogfeld Rauschunterdrückung den sekundären Entrauschungsalgorithmus Anscombe VST und lassen Sie die Modulation recht niedrig, etwa 50-60 %. Sie wollen nur das Rauschen reduzieren, damit Sie die Anzahl der Iterationen etwas erhöhen können, und nicht ein völlig glattes Bild erzeugen.

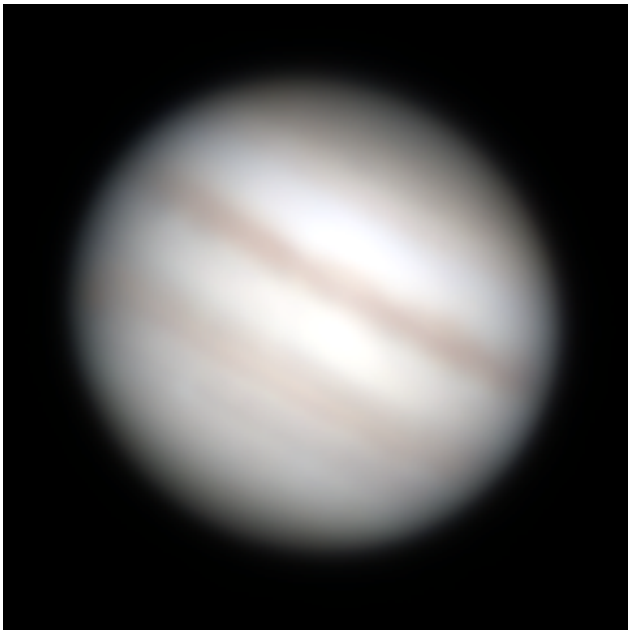
Mondbilder

In der Regel möchten Sie ein Mondbild nach dem Stacking schärfen. Gestackte Mondbilder können mit den Methoden Split Bregman oder Wiener sehr gut geschärft werden. Meine übliche Wahl ist Split Bregman. Versuchen Sie, den Wert von α auf dem Standardwert zu belassen und das Bild mit einer blind geschätzten $_0$ PSF zu entfalten. Das folgende Beispiel zeigt ein frisch gestacktes Mondbild (d.h. es wurde noch keine Wavelet-Verarbeitung durchgeführt).

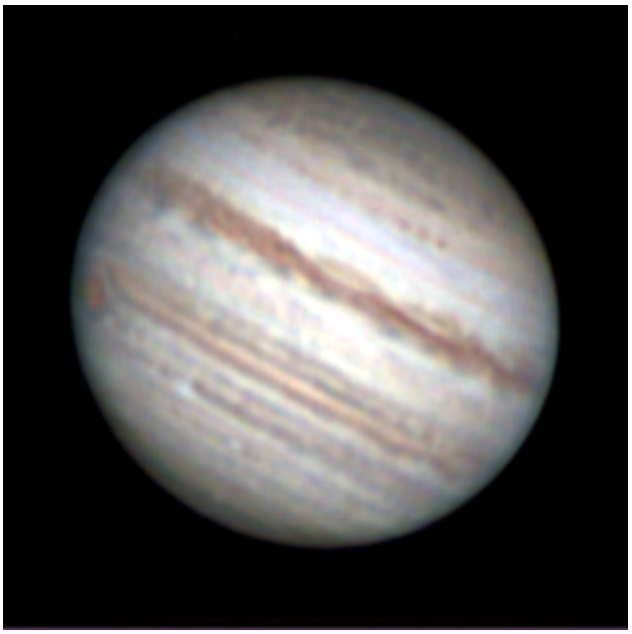
Gestackte Planetenaufnahmen

Ein typische Workflow für Planetenaufnahmen beinhaltet das Stacken des planetarischen SER-Videos in einem Spezialwerkzeug wie Autostakkert! oder Astrosurface und die anschließende Schärfung des resultierenden Bildes mit Hilfe von Wavelets und Dekonvolution. Eine Kombination aus dem Siril-Werkzeug A trous Wavelets und dem Werkzeug Dekonvolution liefert hervorragende Ergebnisse, wie hier gezeigt. Dieses Bild von Jupiter wurde zunächst mit Wavelets geschärft, wobei der erste Ebenenregler auf 75, der zweite auf 10 und alle anderen auf die Standardeinstellungen gesetzt

wurden. Anschließend wurde eine Farb-PSF aus drei Beugungsscheibchen erstellt, die für das verwendete Teleskop und den verwendeten Sensor berechnet wurden (ein 6"-Newton mit einer 3fach-Barlow-Linse und einer ASI462MC-Kamera mit 2,9-Mikrometer-Pixeln), die mit dem RGB-Kompositionswerkzeug zusammengesetzt wurden. Damit wurde das Bild mit 6 Iterationen von Richardson-Lucy (hier habe ich die multiplikative Version verwendet) entfaltet. Mit jedem Schritt wird das Bild schärfer.

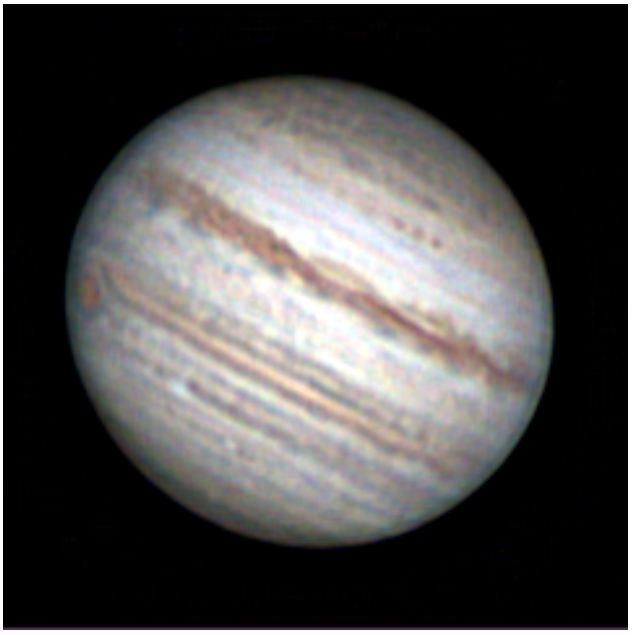


Rohstack, noch unscharf.



Bearbeitet in Siril mit der Wavelet-Zerlegung, Wavelet-Schicht 1 Stärke 75, Wavelet-Schicht 2 Stärke 10.

Bearbeitet mit Wavelets wie oben, und dann mit 6 Iterationen der multiplikativen Richardson-Lucy-Entfaltung geschärft.

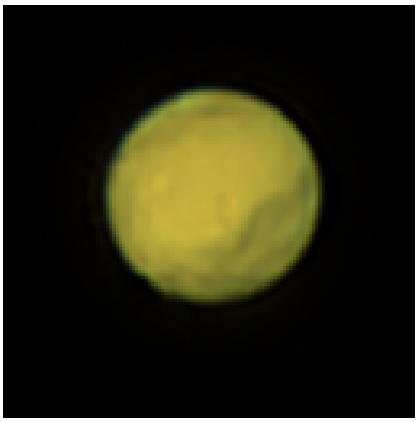


Ungestackte Planetensequenzen

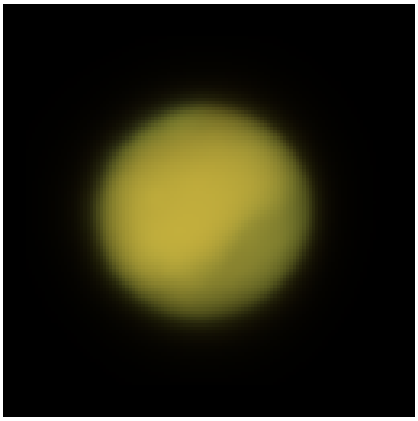
Tipp

Achtung: Diese Methode ist extrem langsam, da sie die individuelle Verarbeitung von typischerweise 30.000 (oder mehr) Bildern in einer Planetensequenz erfordert!

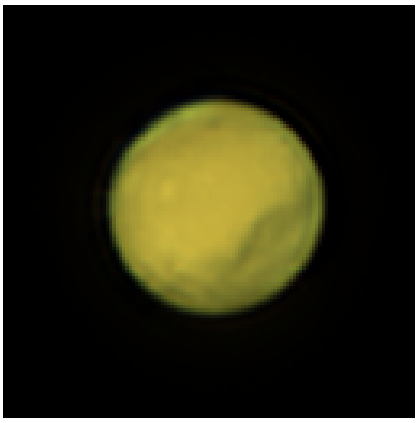
Vielleicht möchten Sie versuchen, die Beugung der Teleskopoptik vor dem Stacking zu korrigieren, indem Sie Ihre Sequenz mit einer Beugungsscheibchen-PSF entfalten. Um dies mit einer typischen One-Shot-Farb-Planetenkamera zu tun, muss die Sequenz beim Laden auf "Debayern" eingestellt werden. Wenn Sie möchten, können Sie noch einen Schritt weiter gehen und drei separate Beugungsscheibchen für rote, grüne und blaue Wellenlängen erzeugen (typischerweise 600 nm, 530 nm bzw. 450 nm). Siril kann nicht direkt eine Farb-PSF erzeugen (die Dekonvolution-Benutzeroberfläche ist komlex genug!), aber wenn Sie jede der roten, grünen und blauen Beugungsscheibchen separat speichern, können Sie sie mit dem RGB-Kompositionswerkzeug zu einer Farb-PSF kombinieren. Wenn Sie diese speichern und ein Farbbild oder eine Farbsequenz laden, wird die PSF in Farbe geladen und jeder Farbkanal wird mit der entsprechenden PSF entfaltet.



Gestapelt und geschärft, ohne einzelne Bilder zu entfalten.



Rohstapel: die besten 30 % von 91k Einzelbildern werden mit Siril entfaltet.



Ergebnis der Schärfung des individuell entfalteten Stacks.

Im obigen Bild ist eine leichte Verbesserung der Kantenform in der Version zu erkennen, die vor dem Stacking Bild für Bild mit einer Beigungsscheibchen-PSF nach der Richardson-Lucy-Methode von Siril entfaltet wurde, aber es ist Vorsicht geboten, um Detailverluste zu vermeiden. Dieser Prozess ist sehr langsam: Meine Entwicklungsmaschine benötigte 4,5 Stunden für die Dekonvolution jedes der 91k Einzelbilder in dieser Sequenz, und die Verbesserung dürfte, wenn überhaupt, nur gering sein.

Befehle

Siril Kommandozeile

```
makepsf clear
makepsf load filename
makepsf save [filename]
makepsf blind [-l0] [-si] [-multiscale] [-lambda=] [-comp=] [-ks=] [-savepsf=]
makepsf stars [-sym] [-ks=] [-savepsf=]
makepsf manual { -gaussian | -moffat | -disc | -airy } [-fwhm=] [-angle=] [-ratio=] [-
↪beta=] [-dia=] [-fl=] [-wl=] [-pixelsize=] [-obstruct=] [-ks=] [-savepsf=]
```

Erzeugt eine PSF zur Verwendung mit der Dekonvolution, eine der drei Methoden die durch die Befehle RL, SB oder WIENER erstellt werden. Als erstes Argument muss eine der folgenden Optionen angegeben werden: **clear**

(löscht die vorhandene PSF), **load** (lädt eine PSF aus einer Datei), **save** (speichert die aktuelle PSF), **blind** (blinde Schätzung der PSF), **stars** (erzeugt eine PSF auf der Grundlage von gemessenen Sternen aus dem Bild) oder **manual** (erzeugt eine PSF manuell auf der Grundlage einer Funktion und von Parametern).

Bei Verwendung des Arguments **clear** sind keine zusätzlichen Argumente erforderlich.

Um ein zuvor gespeichertes PSF zu laden, benötigt das Argument **load** den PSF-*Dateinamen* als zweites Argument. Dieser kann in jedem beliebigen Format vorliegen, für das Siril kompiliert wurde, muss aber quadratisch sein und sollte idealerweise eine ungerade Seitenlänge haben.

Zum Speichern des aktuellen PSF wird das Argument **save** verwendet. Optional kann ein Dateiname angegeben werden (dieser muss eine der Erweiterungen ".fit", ".fits", ".fts" oder ".tif" haben), wird jedoch keiner angegeben, so wird das PSF anhand des Namens der geöffneten Datei oder Sequenz benannt.

Für **blind** können die folgenden optionalen Argumente angegeben werden: **-l0** verwendet die l0-Abstiegsmethode, **-si** verwendet die Methode der spektralen Unregelmäßigkeit, **-multiscale** konfiguriert die l0-Methode für eine PSF-Schätzung mit mehreren Skalierungen, **-lambda=** gibt die Regularisierungskonstante an.

Für PSF von gefundenen Sternen ist der einzige optionale Parameter **-sym**, mit dem die PSF als symmetrisch konfiguriert wird.

Für eine manuelle PSF muss eine der Optionen **-gaussian**, **-moffat**, **-disc** oder **-airy** angegeben werden, um die PSF-Funktion zu spezifizieren. Für Gauß- oder Moffat-PSFs können die optionalen Argumente **-fwhm=**, **-angle=** und **-ratio=** angegeben werden. Für Moffat-PSFs kann auch das optionale Argument **-beta=** angegeben werden. Wenn diese Werte nicht angegeben werden, werden die gleichen Werte wie im Dekonvolutionsdialog verwendet. Für Scheiben-PSFs ist nur das Argument **-fwhm=** erforderlich, das für diese Funktion verwendet wird, um den *Durchmesser* der PSF festzulegen. Für Airy-PSFs (Beugungsscheibchen) können die folgenden Argumente angegeben werden: **-dia=** (legt den Durchmesser der Teleskopöffnung fest), **-fl=** (legt die Teleskopbrennweite fest), **-wl=** (legt die Wellenlänge fest, für die das Airy-Beugungsmuster berechnet werden soll), **-pixelsize=** (legt die Sensorpixelgröße fest), **-obstruct=** (legt die zentrale Obstruktion als Prozentsatz der Gesamtöffnung fest). Wenn diese Parameter nicht angegeben werden, ist die Wellenlänge standardmäßig 525 nm und die zentrale Obstruktion standardmäßig 0 %. Siril versucht, die anderen Parameter aus dem geöffneten Bild auszulesen, aber manche Bildbearbeitungsprogramme stellen möglicherweise nicht alle Parameter zur Verfügung, so dass Sie schlechte Ergebnisse erhalten. Sie werden aus Erfahrung lernen, welche Metadaten Sie bei Ihrer speziellen Bildbearbeitungssoftware getrost weglassen können.

Für jede der oben genannten Optionen zur PSF-Erzeugung kann das optionale Argument **-ks=** angegeben werden, um die PSF-Dimension festzulegen, und das optionale Argument **-savepsf=filename** kann verwendet werden, um das erzeugte PSF zu speichern: ein Dateiname muss angegeben werden, und es gelten dieselben Anforderungen an die Dateinamenerweiterung wie für **makepsf save filename**

Verweise: *psf*, *rl*, *sb*, *wiener*

Siril Kommandozeile

```
rl [-loadpsf=] [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Stellt ein Bild nach der Richardson-Lucy-Methode wieder her.

Optional kann eine PSF mit dem Argument **-loadpsf=filename** (erzeugt mit MAKEPSF) geladen werden.

Die Anzahl der Iterationen wird durch **-iters** festgelegt (der Standardwert ist 10).

Die Art der Regularisierung kann mit **-tv** für die Gesamtvariation oder **-fh** für die Frobenius-Norm der Hessian-Matrix festgelegt werden (die Vorgabe ist keine) und **-alpha=** gibt die Stärke der Regularisierung an (kleinerer Wert = stärkere Regularisierung, Vorgabe = 3000).

Standardmäßig wird die Methode des Gradientenabstiegs mit einer Standardschrittweite von 0,0005 verwendet, doch kann die multiplikative Methode mit **-mul** erzwungen werden.

Das Stopkriterium kann durch Angabe einer Anhaltegrenze mit **-stop=** aktiviert werden

Verweise: *psf*, *makepsf*

Siril Kommandozeile

```
sb [-loadpsf=] [-alpha=] [-iters=]
```

Stellt ein Bild nach der Split-Bregman-Methode wieder her.

Optional kann ein PSF mit dem Argument **-loadpsf=filename** geladen werden.

Die Anzahl der Iterationen wird durch **-iters** angegeben (Standard ist 1).

Der Regularisierungsfaktor **-alpha=** gibt die Stärke der Regularisierung an (kleinerer Wert = stärkere Regularisierung, Standardwert = 3000)

Verweis: *psf*

Siril Kommandozeile

```
wiener [-loadpsf=] [-alpha=]
```

Stellt ein Bild mit Hilfe der Wiener Entfaltungsmethode wieder her.

Optional kann ein PSF (erzeugt mit MAKEPSF) mit dem Argument **-loadpsf=filename** geladen werden.

Der Parameter **-alpha=** gibt den Regularisierungsfaktor für Gaußsches Rauschen an

Verweise: *psf*, *makepsf*

Quellenverzeichnis

10.4.6 Kantenerhaltende Filter

Siril offers two edge preserving filters: OpenCV's implementation of the Bilateral Filter, and a Guided Filter. Both of these filters can be used to reduce noise, and both preserve sharp edges and strong gradients in the image. The bilateral filter acts on a single image whereas the guided filter filters the image using a guide image to identify structures. In the simplest use, the guided filter acts in a self-guided manner by using the input image as the guide image.

Der Aufbau des Fensterdialogs ist recht einfach und es sind nur wenige Einstellungen möglich.

- **Filter type**: Choose between the bilateral filter and the guided filter.
- **Diameter**: This defines the size of the filter kernel that will be used. When using the bilateral filter, setting a diameter of 0 will cause the filter kernel size to be set automatically based on the spatial sigma value. When using the guided filter this value must be set: a diameter of 0 will result in no filtering being done.
- **Sigma (spatial)**: This defines the spatial extent of the filter kernel. A larger value results in smoothing of noise over a greater image area, but is slower to compute. A smaller value results in more local smoothing and is faster to compute. Defaults are not set for the sigma values as the appropriate value may depend significantly on image characteristics, but it can be good to start with both sigma (spatial) and sigma (intensity) set to about 11.
- **Sigma (intensity)**: This defines the range of intensity variation to which the filter responds. A high value results in stronger filtering of noise but may result in loss of genuine detail; a low value reduces the filtering of noise, but will avoid loss of details with gentler gradients.
- **Guide image**: This allows selection of an image to use as a guide image when performing a guided filter. If the "self guided" check box is checked, the filter acts in self-guided mode using the input image as the guide image.
- **Modulation**: In Siril ist die Modulation ein Parameter zwischen 0 und 1, der das Originalbild und das bearbeitete Bild mischt. Bei einem Wert von 1 bleibt nur das verarbeitete Bild erhalten, bei einem Wert von 0 wird überhaupt kein kantenbewahrender Filter angewendet.

Tipp

The parameters to the two different filter types do not behave identically, so when changing between bilateral and guided filters with sigma parameters set, you should expect to see some change in the preview result. The code applies a bit of compensation to the parameters provided to minimize the difference in behaviour of the two filters for the same input parameters but this is not exact (nor is it intended to be).

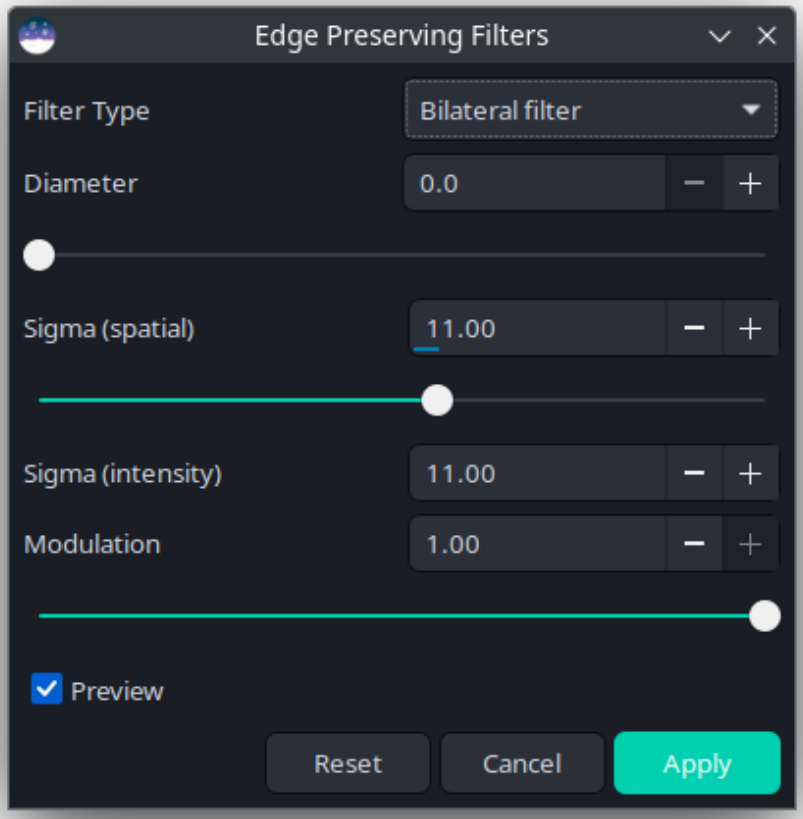


Abb. 48: Dialogbox Median Filter.

Abb. 49: Showing the effect of changing the bilateral filter parameters. Note that the exact values required will depend on the noise characteristics of your data.

Siril Kommandozeile

```
epf [-guided] [-d=] [-si=] [-ss=] [-mod=] [-guideimage=]
```

Wendet einen kantenerhaltenden Filter an. Standardmäßig wird ein bilateraler Filter angewendet; ein geführter Filter kann mit dem Argument **-guided** angegeben werden. Der Filterdurchmesser ist standardmäßig auf 3 eingestellt und kann mit **-d=** festgelegt werden. Seien Sie vorsichtig mit d-Werten größer als 20, da der Algorithmus rechenintensiv sein kann.

Der Sigma-Wert für die Intensitätsfilterung kann mit **-si=** und der räumliche Sigma-Wert mit **-ss=** eingestellt werden. Sigma-Werte stellen die Differenz der Pixelwerte dar, über die der Filter stark wirkt: Für 32-Bit-Bilder sollte der Wert zwischen 0 und 1,0 liegen, während er für 16-Bit-Bilder zwischen 0 und 65535 liegen sollte. Wenn nichts angegeben wird, sind beide Werte standardmäßig auf 11 gesetzt. Wenn **-d=0** gesetzt ist, wird der Filterdurchmesser automatisch basierend auf dem Wert von **-ss** gesetzt. *Beachten Sie, dass beim Anwenden eines geführten Filters nur * *-sc** gilt.*

Bei der Angabe eines geführten Filters kann mit **-guideimage=** ein Führungsbild festgelegt werden. Wenn kein Führungsbild angegeben wird, wird standardmäßig ein selbstgeführter Filter ausgeführt. *Hinweis: Das Führungsbild muss dieselben Abmessungen haben wie das zu filternde Bild!*

Die Stärke des Filters kann mit dem Argument **-mod=** moduliert werden. Bei mod = 1,0 wird die volle Wirkung des Filters angewendet; bei mod kleiner als 1,0 wird ein Teil des Originalbilds mit dem Ergebnis gemischt und bei mod = 0,0 wird keine Filterung angewendet.

10.4.7 Fourier Transformation

Eine Fourier-Transformation (FT) ist eine mathematische Transformation, bei der Funktionen in Frequenzkomponenten zerlegt werden, die durch das Ergebnis der Transformation als Funktion der Frequenz dargestellt werden. Diese Transformation wird häufig in der Bildgebung verwendet, da sie es ermöglicht, Signale mit regelmäßigen Frequenzen zu sehen.

Theorie

Fourier Transformation

Die Fourier-Transformation ist ein Analyseverfahren, bei dem eine komplexwertige Funktion $f(x)$ in ihre Bestandteile und deren Amplituden zerlegt wird:

$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \xi x}\, dx.$

Inverse Transformation

Der inverse Transformation ist die Synthese, die $f(x)$ aus seiner Transformation reproduziert:

$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x}\, d\xi.$

Siril ermöglicht die Umwandlung eines Bildes in den Frequenzraum mit Hilfe eines Algorithmus zur schnellen Fourier-Transformation. Das Ergebnis liegt in Form von zwei Bildern vor. Das erste Bild, das automatisch geladen wird, enthält den Betrag (oder Modulus) der Transformation, das zweite Bild enthält die Phase. Die Namen der beiden Bilder müssen auf der Registerkarte **Direkttransformation** (siehe Abbildung unten) des Dialogs eingegeben werden. Es ist dann

möglich, das Modulusbild zu verändern, indem Frequenzspitzen, die unerwünschten Signalen entsprechen, entfernt werden. Es ist wichtig, dass Sie nicht vergessen, die Änderungen zu speichern.

Wenn die Option *Zentriert* aktiviert ist, wird der Ursprung der direkten Fourier-Transformation zentriert. Ist sie nicht aktiviert, befindet sich der Ursprung in der linken, oberen Ecke.

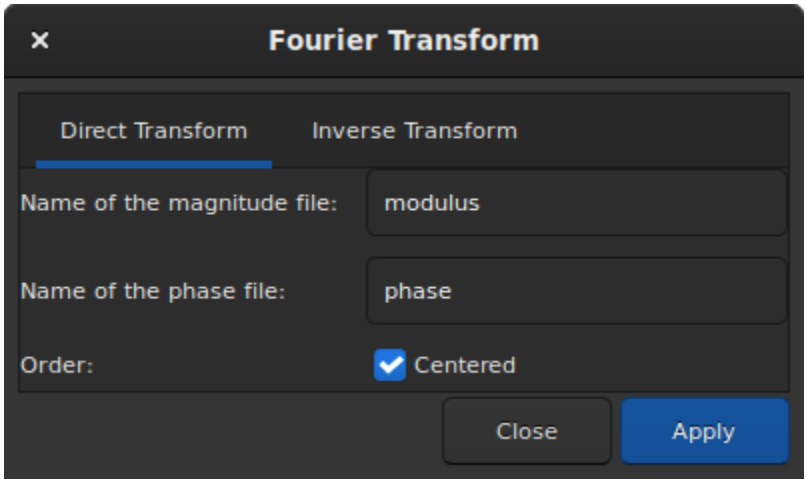


Abb. 50: Registerkarte Direkte Transformation.

Um das Bild zu rekonstruieren, klicken Sie auf die Registerkarte **Inverse Transformation** und geben Sie den Dateipfad der Modul- und Phasenbilder ein.

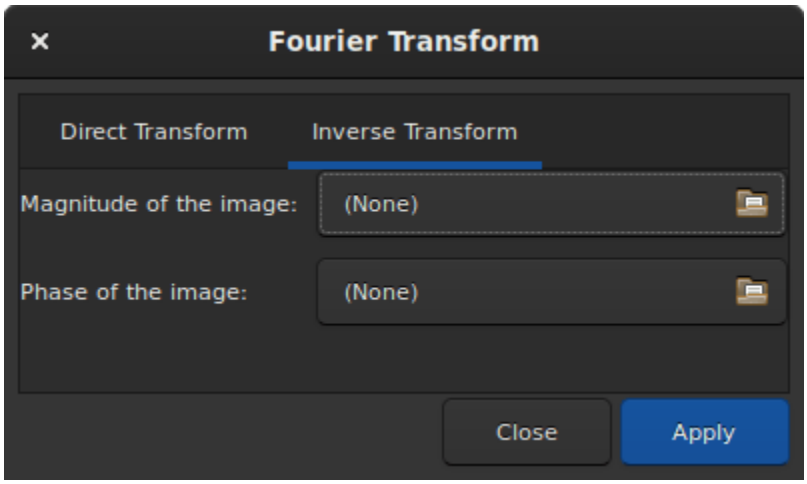


Abb. 51: Registerkarte Inverse Transformation.

Siril Kommandozeile

```
fftd modulus phase
```

Wendet eine Fast-Fourier-Transformation auf das geladene Bild an. **modulus** und **phase**, die im Argument angegeben sind, sind die Dateinamen der gespeicherten FITS-Dateien

Siril Kommandozeile

```
ffti modulus phase
```

Ermittelt ein korrigiertes Bild unter Anwendung einer inversen Transformation. Als **nodulus** und **phase** werden die Dateien verwendet, die im Argument angegeben werden. Das Ergebnis wird als neues geladenes Bild angezeigt

10.4.8 Median-Filter

Der Median stellt den mittleren Datenpunkt dar, wobei die Hälfte der Daten kleiner und die Hälfte der Daten größer als dieser Punkt ist. Dies ist ein robuster Schätzerwert, um Ausreißer aus einem Datensatz zu entfernen. Folglich kann dieses Werkzeug als naiver Entrauscher nützlich sein, der gegen *Impulsrauschen* wirksam ist.

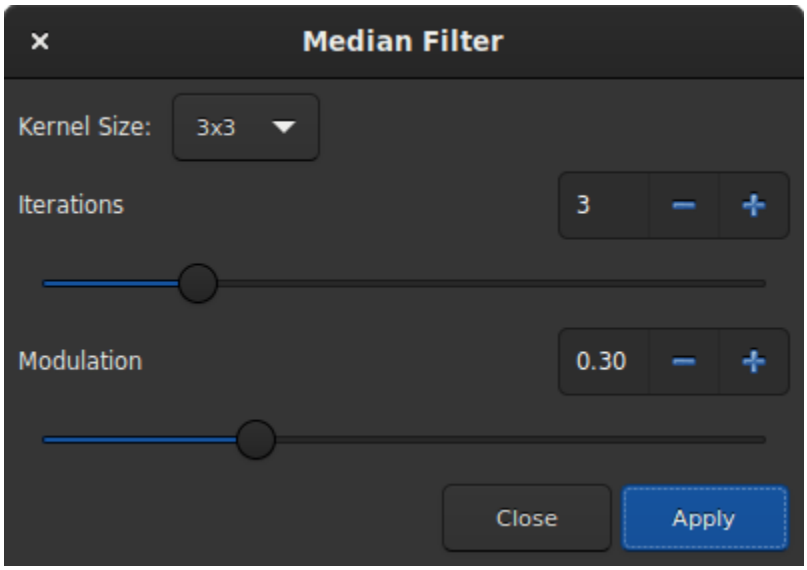


Abb. 52: Dialogbox Median Filter.

Der Aufbau des Fensterdialogs ist recht einfach und es sind nur wenige Einstellungen möglich.

- **Kernelgröße**: Von $3imes3$ bis $15imes15$ wird hier die Größe des quadratischen Kerns festgelegt, der zur Anwendung des Filters verwendet wird. Je größer der Kernel ist, desto unschärfer wird das Ergebnis sein.
- **Iterationen**: Hier wird die Anzahl der Durchläufe des Kernels festgelegt.
- **Modulation**: In Siril ist die Modulation ein Parameter zwischen 0 und 1, der das Originalbild und das bearbeitete Bild mischt. Bei einem Wert von 1 bleibt nur das verarbeitete Bild erhalten, bei einem Wert von 0 wird überhaupt kein Medianfilter angewendet.

Siril Kommandozeile

```
fmedian ksize modulation
```

Führt einen Medianfilter der Größe **ksize** x **ksize** (**ksize** MUSS ungerade sein) auf das geladene Bild mit einem Modulationsparameter **modulation** aus.

Das Ausgabepixel wird wie folgt berechnet: out=mod x m + (1 - mod) x in, wobei m der Median-gefilterte Pixelwert ist. Bei einem Modulationswert von 1 wird keine Modulation angewendet

10.4.9 Rauschreduzierung

Bildrauschen

Bilder leiden unter verschiedenen Arten von Rauschen:

1. Impulsrauschen
   - Diese Art von Rauschen (manchmal auch "Salz- und Pfefferrauschen" genannt) entsteht in der Regel durch Hot- oder Coldpixel. In der Regel wird es durch die Verwendung eines Sigma-Rejection-Stacking behandelt, aber manchmal müssen Sie es auch manuell behandeln, wenn Sie ein einzelnes, nicht gestacktes Bild verarbeiten.
2. Additives weißes Gaußsches Rauschen
   - Diese Art von Rauschen ist typisch für stark belichtete Fotos: Es entsteht durch die thermischen und elektronischen Schwankungen des Aufnahmegeräts, und der Rauschpegel ist unabhängig vom Signal. Additives weißes Gaußsches Rauschen (Additive White Gaussian Noise - AWGN) kann während der Aufnahme durch die Verwendung gekühlter Kameras reduziert werden, und es wird beim Stacking reduziert, weil das Stapeln von n Bildern das korrelierte Signal um einen Faktor von n erhöht, während das unkorrelierte Rauschen nur um einen Faktor von $\sqrt{n}$ zunimmt. Dies ist auch die Art von Rauschen, auf deren Entfernung die meisten klassischen Entrauschungsalgorithmen ausgelegt sind.
3. Poissonrauschen
   - Bei photonenarmen Bildern ist das Rauschen nicht mehr in erster Linie gaußförmig, sondern die Wahrscheinlichkeitsverteilung der Photonenzählung wird signifikant oder sogar dominant. Dies wird durch eine Poisson-Verteilung modelliert, und diese Art von Rauschen ist signalabhängig.

Rauschreduzierung in Siril

Siril bietet gut untersuchte klassische Entrauschungsalgorithmen auf dem neuesten Stand der Technik. Die Kriterien für die Auswahl der Algorithmen waren:

- Der Algorithmus sollte in wissenschaftlichen Fachzeitschriften mittels Peer Review analysiert werden, mit einer Beschreibung des Algorithmus und einem objektiven quantitativen Vergleich seiner Leistung.
- Die Autoren sollten eine F/OSS-Implementierung zur Verfügung gestellt haben. Dies ist wichtig, um IP-Probleme zu vermeiden und, wenn die Referenzimplementierungen direkt verwendet wurden, die Lizenzkompatibilität sicherzustellen.
- Die Algorithmen sollten mit einer angemessenen Geschwindigkeit arbeiten.
- Schließlich muss die Implementierung des Algorithmus in der Lage sein, 32-Bit-Gleitkomma-Pixeldaten zu verarbeiten.

Die Technologie der neuronalen Netze zur Rauschunterdrückung wurde untersucht, aber zum gegenwärtigen Zeitpunkt aufgrund der Komplexität der Entwicklung verworfen. Die Entrauschungsleistung neuronaler Netze kann klassische Ansätze in der Regel um bis zu einem dB Peak des Signal-Rausch-Verhältnisses übertreffen, aber die Leistung hängt in hohem Maße davon ab, dass das neuronale Netz auf Daten trainiert wird, die für die Live-Daten repräsentativ sind auf denen es angewendet werden soll.

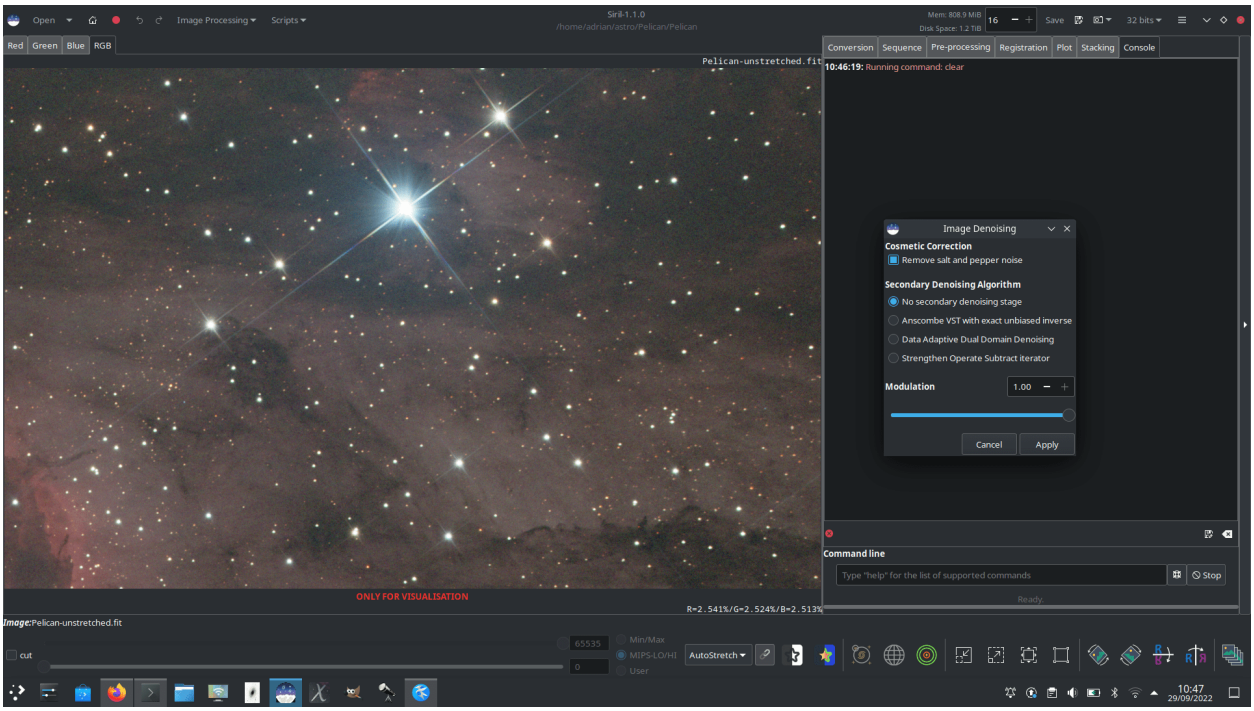


Abb. 53: Dialog Bildentrauschung

Algorithmen: Impulsrauschen

Siril entfernt das Impulsrauschen in erster Linie durch Stacking mit Sigma-Unterdrückung. Wenn Sie diese Stacking-Methode verwenden, sollten Sie keine Probleme mit Impulsrauschen haben. Wenn Sie jedoch an einer Einzelbelichtung arbeiten, können Sie durchaus Impulsrauschen in Ihrem Bild finden. Dieses sollte mit der Funktion **Kosmetische Korrektur** von Siril behandelt werden, bevor eine andere Rauschreduzierung verwendet wird, da das Vorhandensein von Impulsrauschen Entrauschungsalgorithmen für "Additives weißes gaußsches Rauschen" verfälschen und Artefakte erzeugen kann. Die Funktion funktioniert ähnlich wie die Sigma-Rückweisung, allerdings für benachbarte Pixel. Jedes Pixel, dessen Intensität um mehr als n Standardabweichungen von seinen Nachbarn abweicht, wird zurückgewiesen und durch einen Wert ersetzt, der auf dem Median der Nachbarn basiert. Im Entrauschungswerkzeug ist die kosmetische Korrektur standardmäßig aktiviert und findet vor allen weiteren Entrauschungsschritten statt. (Auch wenn bereits eine Impulsrauschen-Entfernung durchgeführt wurde, schadet es nicht, die Einstellung aktiviert zu lassen). Alternativ kann die kosmetische Korrektur auch manuell mit dem Werkzeug *Kosmetische Korrektur* im Menü *Bildbearbeitung* durchgeführt werden.

Algorithmen: Additives weißes Gaußsches Rauschen AWGN

Der wichtigste Algorithmus zur Reduzierung des AWGN-Rauschens, der in Siril verwendet wird, ist die nicht-lokale Bayes'sche Entrauschung (NL-Bayes) [Lebrun2013].

- Nichtlokale Entrauschungsalgorithmen stellen eine wesentliche Verbesserung gegenüber früheren pixelzentrierten linearen Filtern dar. NL-Bayes ist eine verbesserte Version eines früheren nichtlokalen Entrauschungsalgorithmen und bietet einen der besten klassischen AWGN-Entrauschungsalgorithmen. Er ist geringfügig besser als der moderne "Benchmark"-Algorithmus Block Matching and 3D tranform (BM3D) zur Rauschunterdrückung und viel schneller in der Ausführung.
- Der Schlüsselparameter, der zur Optimierung der Leistung von AWGN-Algorithmen erforderlich ist, ist sigma, die Standardabweichung des Rauschens. Siril misst den Rauschpegel direkt aus den Bilddaten und gibt ihn an den NL-Bayes-Algorithmus weiter. Daher gibt es im Siril-Entrauschungstool keine konfigurierbaren Eingaben für NL-Bayes.

Siril ergänzt NL-Bayes mit einer Reihe von anderen Algorithmen zur Rauschunterdrückung:

- Dataadaptive Dual-Domain-Entrauschung (DA3D) [Pierazzo2017]
  - Dabei wird das Ergebnis von NL-Bayes als Vorgabe verwendet. Diese Vorgabe wird zur erneuten Verarbeitung des Originalbildes verwendet, indem eine Schrumpfung im Frequenzbereich an form- und datenadaptiven Feldern vorgenommen wird. Dies verbessert die Leistung von NL-Bayes geringfügig, verursacht aber zusätzliche Rechenkosten. Die form- und datenadaptiven Patches werden dynamisch ausgewählt, so dass sich die Berechnungen auf die Bereiche mit den größten Bilddetails konzentrieren. Sie kann auch dazu beitragen, Treppenartefakte im Originalbild zu reduzieren.
  - Im Siril-Entrauschungstool ist DA3D ein einfacher Schalter ohne optionale Einstellungen.
- Iterator "Operate Subtract" stärken (SOS) [Romano2015]
  - SOS funktioniert, indem der primäre Entrauschungsalgorithmus mehrmals wiederholt wird. Bei jeder Iteration wird das Bild "verstärkt", indem ein Teil des ursprünglichen verrauschten Bildes beigemischt wird. Der NL-Bayes-Algorithmus wird auf diesem verstärkten Bild ausgeführt, woraufhin die vorherige Schätzung subtrahiert wird.
  - Das Bild x in einer Iteration $k+1$ ist gegeben durch $x_{k+1} = f(y + x_k) - x_k$, wobei y das verrauschte Eingangsbild ist.
  - Im Siril-Entrauschungsprogramm ist SOS ein Schalter mit zwei Parametern: Die Anzahl der `SOS Iterationen` kann eingestellt werden, und der Anteil des verrauschten Bildes, der bei jeder Iteration beigemischt wird (`rho`), kann eingestellt werden. Vermeiden Sie es, `rho` zu hoch einzustellen, da dies zu Problemen mit der Konvergenz von SOS führen kann: die Standardwerte (`3 Iterationen` und `rho = 0,2`) sind normalerweise in Ordnung.

Algorithmen: Poisson und Poisson-Gaußsches Rauschen

- Anscombe VST mit exakter unverzerrter Invertierung (Anscombe Variance-Stabilising Transform) [Mäkitalo2011], [Mäkitalo2012]
  - Varianzstabilisierende Transformationen werden für Bilder mit Poisson- oder Poisson-Gaußschem Rauschen verwendet, um die Signalabhängigkeit des Rauschens zu minimieren und es eher wie AWGN aussehen zu lassen, das mit NL-Bayes gut entfernt werden kann, und anschließend wird eine inverse Transformation angewendet. Die für Siril gewählte Transformation ist die Anscombe-Transformation $A : x \rightarrow 2 \times \sqrt{(x + \frac{3}{8})}$
  - Da die Transformation nicht linear ist, würde die Verwendung der direkten algebraischen Umkehrung zu einer Verzerrung des Ergebnisses führen. Siril verwendet daher eine geschlossene Annäherung an die exakte unverzerrte Umkehrung, die schnell zu berechnen ist und eine wesentliche Verbesserung gegenüber anderen Formen der Umkehrung wie der asymptotischen Umkehrung darstellt.
  - Im Siril-Entrauschungstool ist Anscombe VST ein einfacher Schalter ohne optionale Einstellungen.

Beachten Sie, dass nur einer der oben genannten komplementären Entrauschungsalgorithmen gewählt werden kann.

Die nachstehende Animation zeigt, was mit Hilfe der Varianzstabilisierung bei einem photonenarmen Bild möglich ist, in diesem Fall ein einzelnes 5-minütiges Rotfilter-Light des Pelikannebels, das mit der Bildschirmübertragungsfunktion *Auto-Stretch* angezeigt wird. Man beachte das Fehlen von Unschärfen, Aufblähungen oder Detailverlusten um die Sterne herum und die scharfe Kante des Nebels im unteren linken Teil des Bildes im Vergleich zu dem, was man mit einfacheren Rauschunterdrückungsprogrammen erreichen könnte. Wenn man das Bild auf sympathischere Weise streckt und mit anderen Kanälen kombiniert, würde dies die Qualität, die mit sehr begrenzten Daten erreicht werden kann, erheblich verbessern (obwohl mehr Daten immer die bessere Lösung sind!)

Abb. 54: Entrauschen eines photonenarmen Bildes

Modulation

In Siril ist die Modulation ein Parameter zwischen 0 und 1, der das Originalbild und das entrauschte Bild mischt. Bei einem Wert von 1 bleibt nur das entrauschte Bild erhalten, bei einem Wert von 0 wird überhaupt keine Rauschunterdrückung vorgenommen. Die Modulation verringert natürlich die Leistung der Rauschunterdrückung, aber in einigen Fällen, wenn die Rauschunterdrückung flache Bereiche des Bildes etwas zu glatt aussehen lässt, können Sie die Modulation verwenden, um das Aussehen der Mikrotextur in diesen Regionen zu erhalten.

Wann die Rauschunterdrückung eingesetzt werden sollte

Die Algorithmen zur Rauschunterdrückung sind darauf ausgelegt, AWGN zu entfernen, und sollten daher am besten bei ungestreckten Bildern funktionieren: Wenn weißes Rauschen einer nichtlinearen Streckung unterzogen wird, ändern sich seine Eigenschaften und es ist nicht mehr weiß. Die Rauschunterdrückung kann auch bei gestreckten Bildern durchgeführt werden und führt zu einer Verbesserung, ist aber möglicherweise nicht so effektiv wie bei der Anwendung im linearen Bild.

Rauschunterdrückung Benutzeroberfläche

Auf das Siril-Rauschunterdrückungsprogramm kann auf zwei Arten zugegriffen werden: über die grafische Benutzeroberfläche oder über die Befehlsschnittstelle. Die grafische Benutzeroberfläche ist unten abgebildet. Hinweis: Die erweiterten SOS-Optionen sind ausgeblendet, wenn SOS nicht ausgewählt ist.

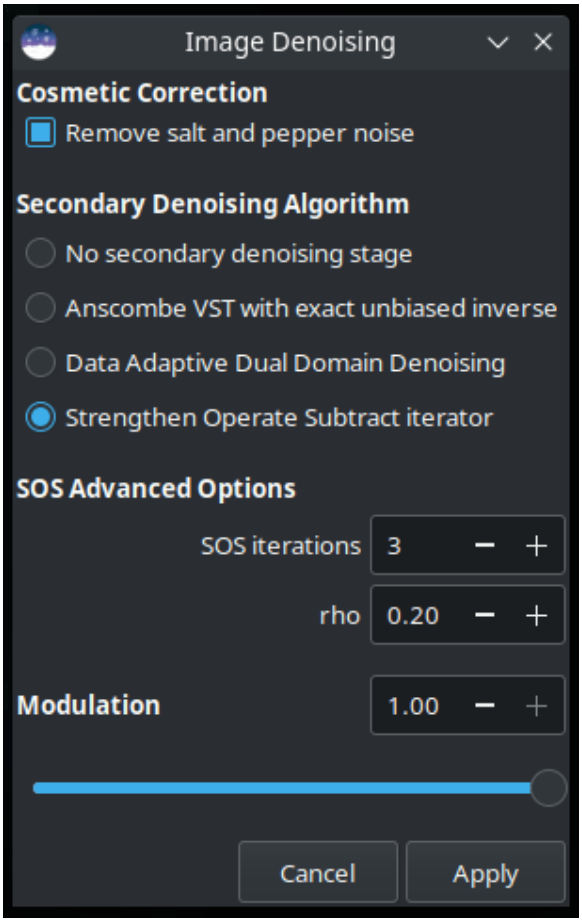


Abb. 55: Siril Bildentrauschungs-GUI

Die Rauschunterdrückung kann auch mit Siril-Befehlen entweder in der Konsole oder in Skripten durchgeführt werden. Das Format ist:

Siril Kommandozeile

```
denoise [-nocosmetic] [-mod=m] [ -vst | -da3d | -sos=n [-rho=r] ] [-indep]
```

Entrauscht dsa Bild mit dem von *Lebrun, Buades und Morel <https://www.ipol.im/pub/art/2013/16>* beschriebenen Algorithmus.

Es wird dringend empfohlen, vor der Entrauschung eine kosmetische Korrektur auszuführen, um Salz- und Pfefferrauschen zu entfernen. Wenn die jedoch bereits zu einem früheren Zeitpunkt im Arbeitsablauf erfolgt ist, kann es hier mit dem optionalen Befehl **-nocosmetic** deaktiviert werden.

Ein optionales Argument **-mod=m** kann angegeben werden, wobei 0 <= m <= 1. Das Ausgabepixel wird dann *out=m x d + (1 - m) x in* berechnet, wobei *d* der entrauschte Pixelwert ist. Bei einem Modulationswert von 1 wird keine Modulation angewendet. Wird der Parameter nicht angegeben, wird er standardmäßig auf 1 gesetzt.

Das optionale argument **-vst** kann verwendet werden, um die verallgemeinerte varianzstabilisierende Anscombe-Transformation (Generalised Anscombe Variance Stabilizing Transform) vor NL-Bayer anzuwenden. Dies ist nützlich bei photonenarmen Bildern, wie z.B. Einzelbildern, bei denen das Rauschen einer Poisson- oder Poisson-Gauß-Verteilung folgt und nicht primär gaußförmig ist. Es kann nicht in Verbindung mit DA3D oder SOS verwendet werden, und für die Entrauschung von gestackten Bildern ist es in der Regel nicht von Vorteil.

Mit dem ptionalen Argument **-da3d** kann die datenadaptive Dual-Domain-Entrauschung (DA3D) als letzter Entrauschungsalgorithmus aktiviert werden. Dabei wird die Ausgabe von BM3D als Vorgabe zur Verfeinerung der Rauschunterdrückung verwendet. Das verbessert die Detailwiedergabe und reduziert Treppenartefakte.

Das optionale Argument **-sio=n** kann verwendet werden, um das iterative Denoise-Boosting nach dem SOS-Prinzip (Strengthen-Operate-Subtract) zu aktivieren, wobei die Anzahl der Iterationen durch n festgelegt wird. Dieser Booster kann insbesondere dann bessere Ergebnisse liefern, wenn der NL-Bayes-Algorithmus ohne Booster Artefakte im Hintergrundbereich erzeugt. Wenn sowohl -da3d als auch -sos=n angegeben sind, gilt die zuletzt angegebene Option.

Das optionale Argument **-rho=r** kann angegeben werden, wobei 0 < r < 1. Dies wird vom SOS-Booster verwendet, um den Anteil des verrauschten Bildes zu bestimmen, der dem Zwischenergebnis zwischen den einzelnen Iterationen hinzugefügt wird. Wenn -sos=n nicht angegeben wird, wird der Parameter ignoriert.

Standardmäßig werden DA3D oder SOS nicht angewendet, da die Verbesserung der Rauschunterdrückung in der Regel relativ gering ist und diese Techniken zusätzliche Verarbeitungszeit erfordern.

In sehr seltenen Fällen können bei der Entrauschung von Farbbildern blockartige Farbartefakte in der Ausgabe entstehen. Das optionale Argument **-indep** kann verwendet werden, um dies zu verhindern, indem jeder Kanal separat entrauscht wird. Dies ist zwar langsamer, beseitigt aber die Artefakte

Vergleich

Die folgenden Bilder zeigen einen einfachen Vergleich der verschiedenen Algorithmen. Beachten Sie, dass nur ein Bild verwendet wird: In der Praxis werden sich verschiedene Algorithmen besser für die Verwendung bei unterschiedlichen Bildern eignen. Alle Bilder können angeklickt werden, um sie mit 100 % Zoom zu betrachten.

Verrauschtes Originalbild

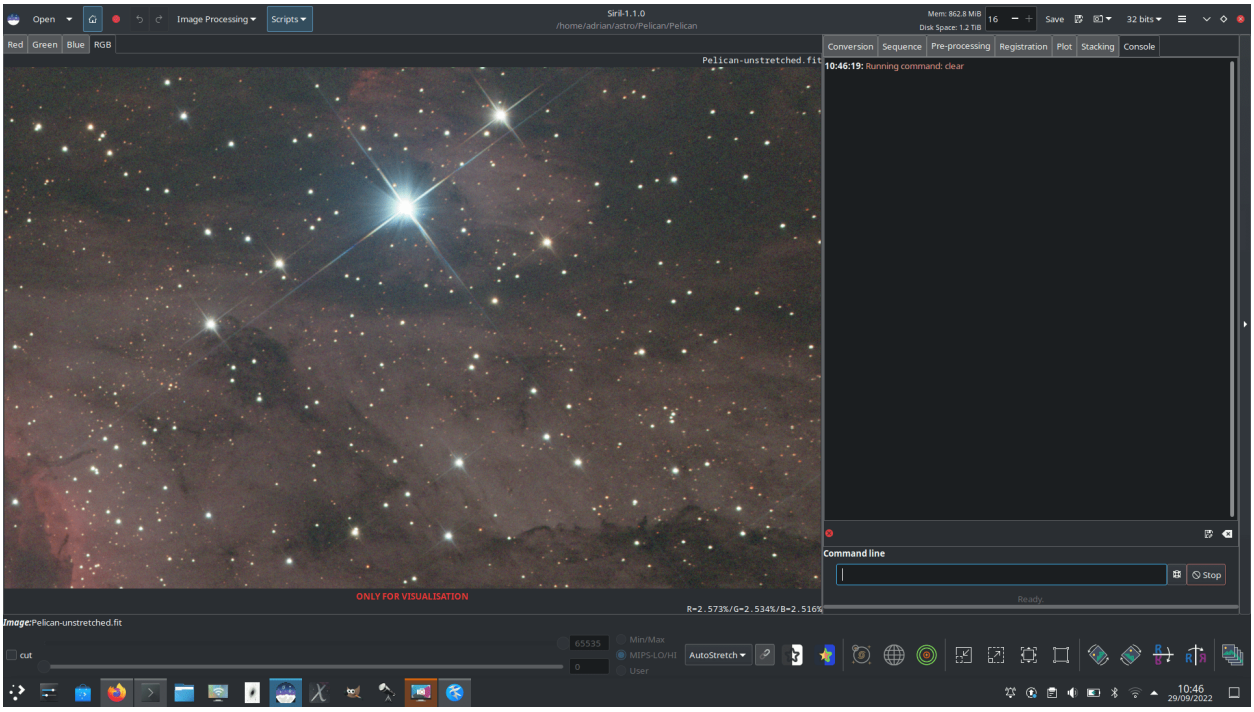


Abb. 56: Verrauschtes Bild

Nur mit NL-Bayes entrauscht

Nur mit NL-Bayes entrauscht, mit 75% Modulation um einen Teil der Mikrotexturen zu erhalten

Entrauscht mit NL-Bayes unter Verwendung der Anscombe VST

Entrauscht mit DA3D unter Verwendung eines NL-Bayes-Führungsbildes

Entrauscht mit NL-Bayes und SOS

Einschränkungen

Die wichtigste Einschränkung besteht darin, dass die Algorithmen am besten funktionieren, wenn das Rauschen gaußförmig ist (oder mit Hilfe des VST annähernd gaußförmig gemacht werden kann). Es gibt einige Gründe, warum dies nicht der Fall sein könnte:

- Wenn das Bild bereits stark bearbeitet wurde, z. B. durch Entfaltung oder Wavelet-Schärfung, ist der Charakter des Rauschens im Allgemeinen nicht mehr gaußförmig. Wenn sowohl Rauschunterdrückung als auch Entfaltung Teil Ihres Arbeitsablaufs sind, sollte die Rauschunterdrückung zuerst durchgeführt werden.
- OSC-Bilder lassen sich weniger gut entrauschen als Mono- oder zusammengesetzte Farbbilder. Es wird zwar eine geringfügige Verringerung des AWGN in der Luminanz erreicht, aber durch das De-Bayering wird der Charakter des Rauschens so verändert, dass es nicht mehr gut als AWGN modelliert und nicht sehr effektiv entfernt wird. Darüber hinaus wird Chrominanzrauschen sowohl bei OSC- als auch bei zusammengesetzten Monofarbbildern in der Regel nicht gut als AWGN modelliert und muss anders behandelt werden. Gegenwärtig wird Chrominanzrauschen am besten mit allgemeiner Bildbearbeitungssoftware wie The GIMP behandelt.

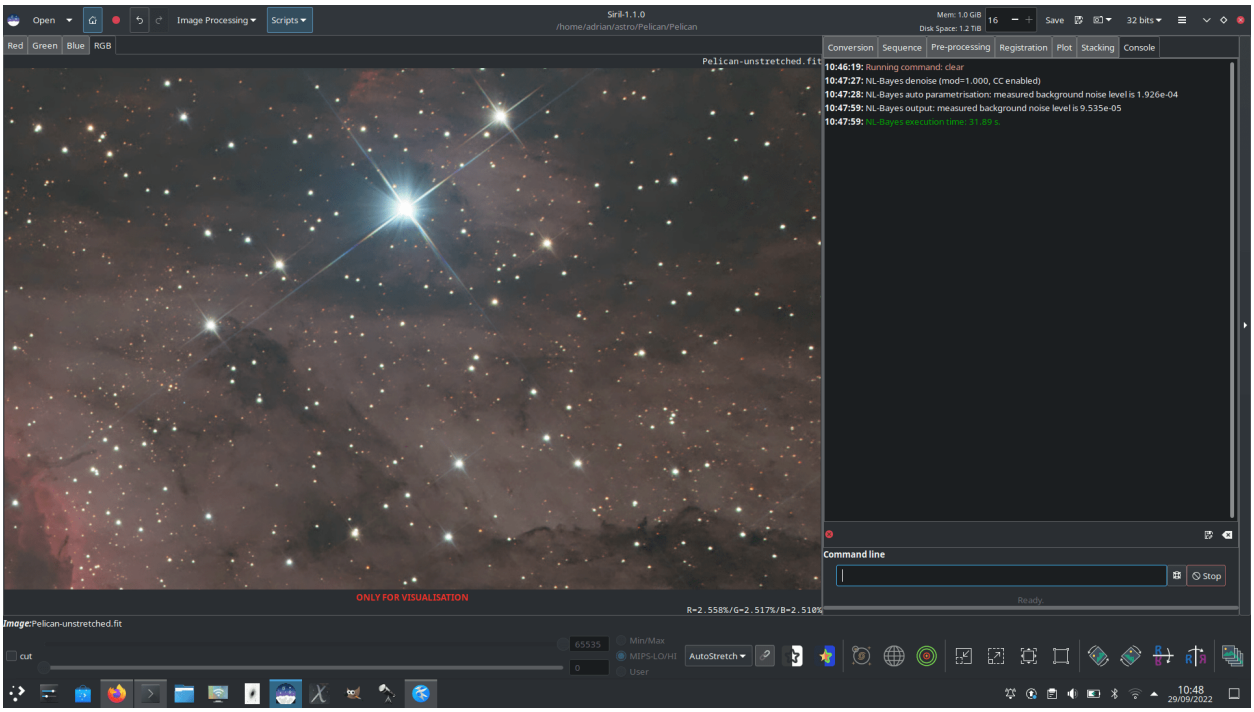


Abb. 57: Nur mit NL-Bayes entrauscht

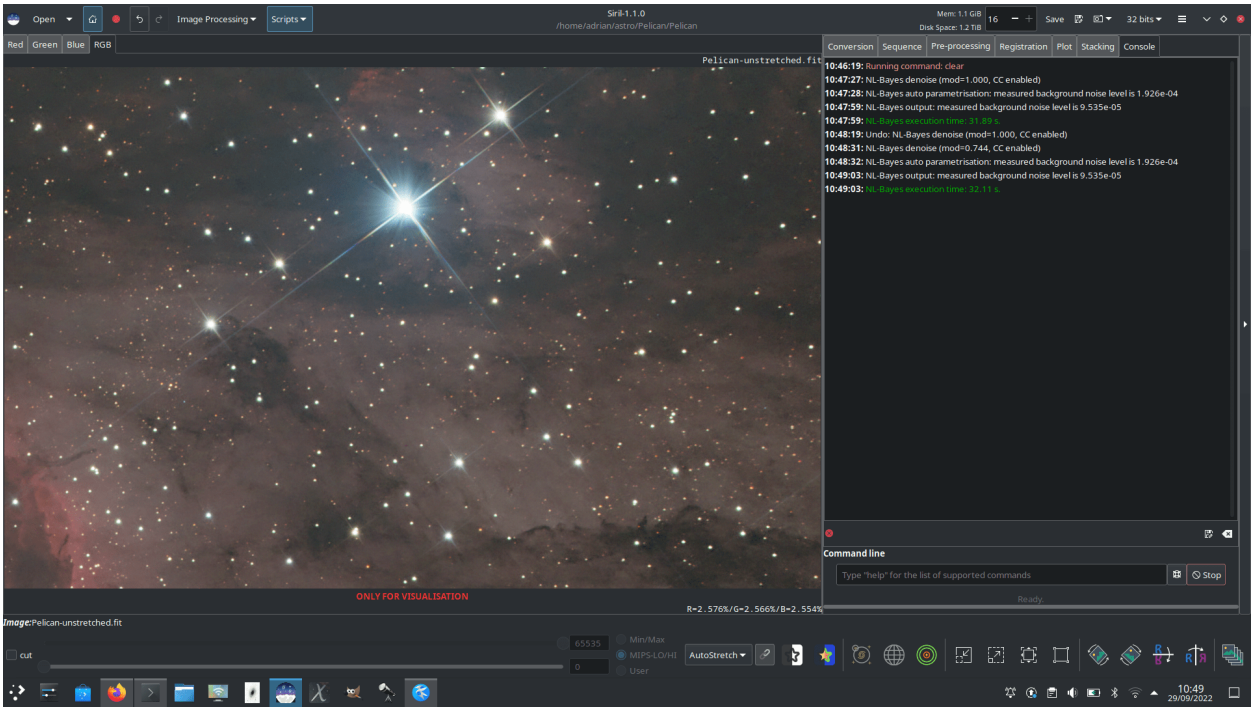


Abb. 58: Verwendung der Modulation

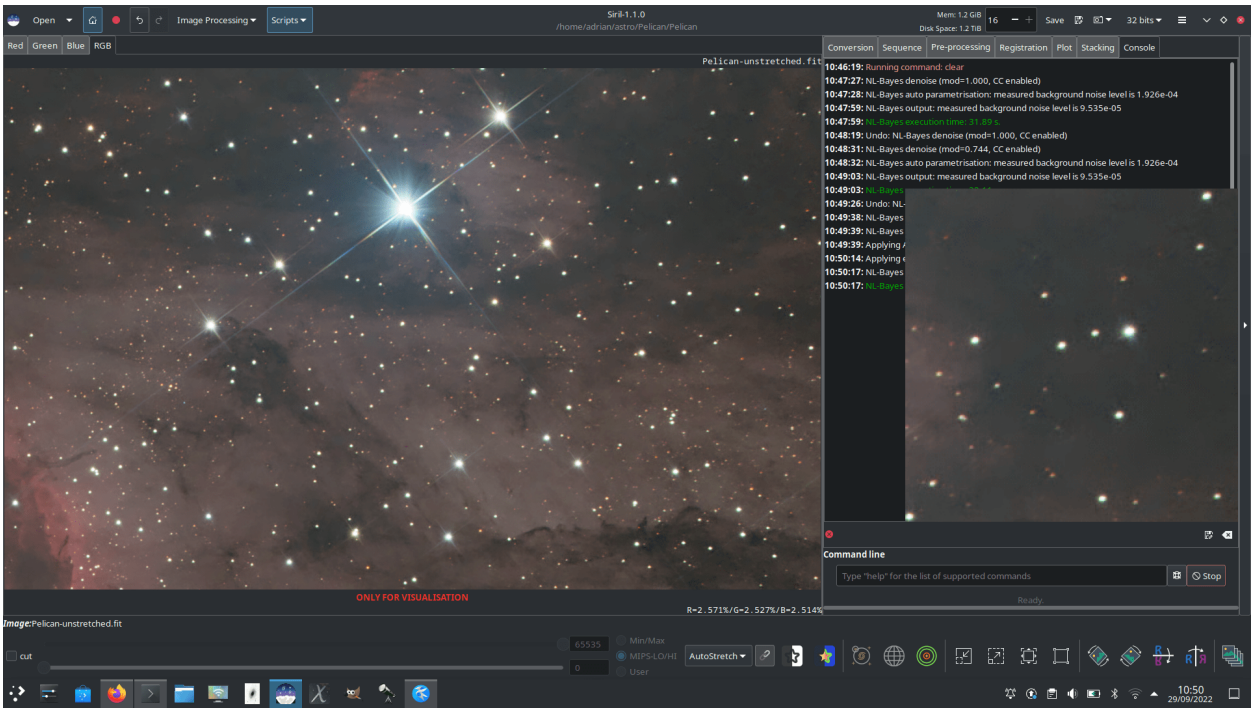


Abb. 59: Entrauscht mit NL-Bayes, Varianzstabilisierung mit Anscombe-Transformation. Rechts ist ein nicht interpolierter 200%-Zoom zu sehen.

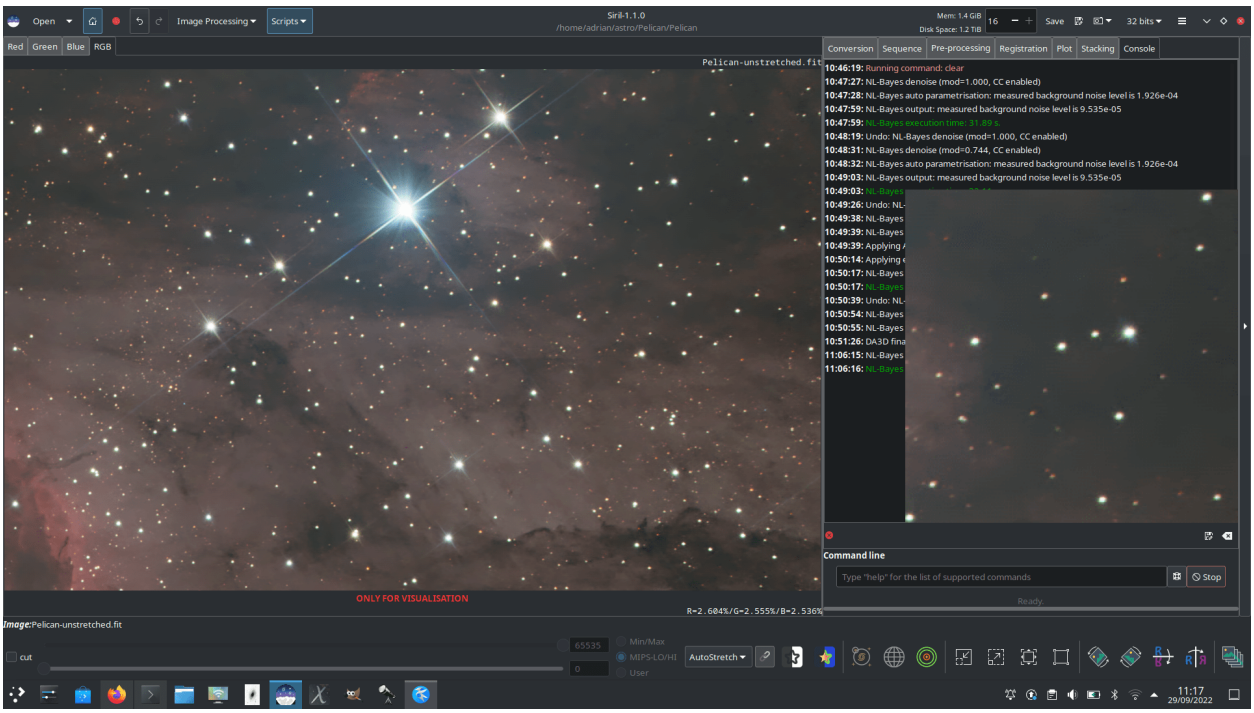


Abb. 60: Mit DA3D entrauschtes, mit NL-Bayes aufbereitetes Führungsbild. Rechts ist ein nicht interpolierter 200%-Zoom zu sehen.

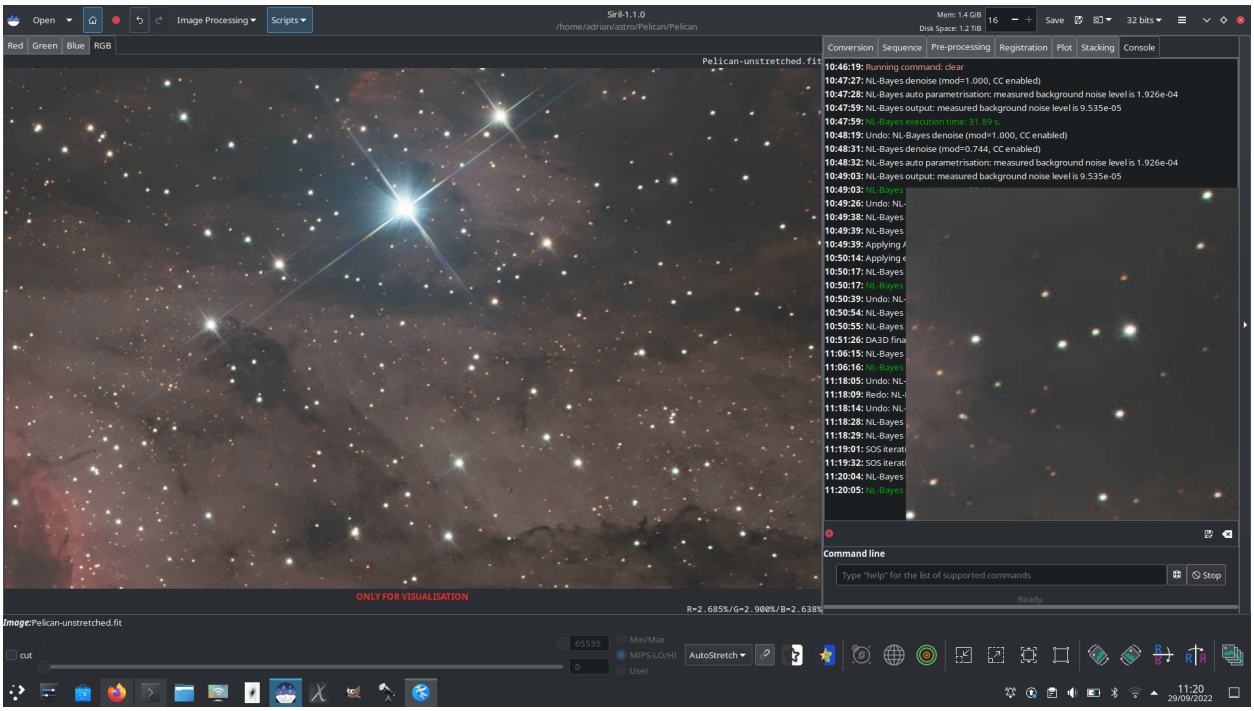


Abb. 61: Entrauscht mit NL-Bayes und SOS-Iterationen. Rechts ist ein nicht interpolierter 200%-Zoom zu sehen.

Quellenverzeichnis

10.4.10 Larson Sekanina Filter (Rotationsgradient)

Der Rotationsgradient, auch Larson Sekanina filter genannt, ist ein Filter, mit dem kreisförmige Strukturen aus einem Bild entfernt werden können, um andere Details besser hervorzuheben. Diese Technik ist besonders effektiv, um die aus dem Kern eines Kometen austretenden Strahlen zu zeigen.

Das Prinzip ist ganz einfach: Diese Bildbearbeitung besteht darin, zwei Kopien des Bildes voneinander zu subtrahieren, wobei eine der beiden Kopien zuvor in Bezug auf einen im Bild definierten Punkt gedreht worden ist.

- Wenn um diesen Punkt herum kreisförmige Strukturen vorhanden sind, werden diese durch die Drehung nicht verändert und verschwinden nach der Drehung.
- Wenn es nicht kreisförmige Strukturen gibt, wie z. B. Jets in der Koma, werden diese zwischen den beiden Kopien gegeneinander verschoben, und die Subtraktion verstärkt den Kontrast dieser Strukturen im Ergebnis.
- Wenn sich der Komet im Bild bewegt, ist es möglich, eine radiale Verschiebung hinzuzufügen.

Theorie

Ausgehend von einem Eingabebild erzeugt der Filter zwei Bilder, jedes mit einer radialen Verschiebung (dr in Pixeln) und einer Rotationsverschiebung ($d\alpha$ in Grad) relativ zum Punkt (x_c, y_c). Diese Verschiebungen haben in beiden Bildern dieselbe Größe, aber entgegengesetzte Vorzeichen. Die beiden Bilder werden dann kombiniert, um das endgültige Bild zu erzeugen. In Polarkoordinaten (r, a) in Bezug auf den Punkt (x, y) haben wir:

$$B'(\alpha, r, d\alpha, dr) = 2 \cdot B(\alpha, r) - B(\alpha - d\alpha, r - dr) - B(\alpha + d\alpha, r - dr)$$

mit B: dem Startbild, B': dem Ergebnisbild, dr: der radialen Verschiebung und $d\alpha$: der Rotationsverschiebung

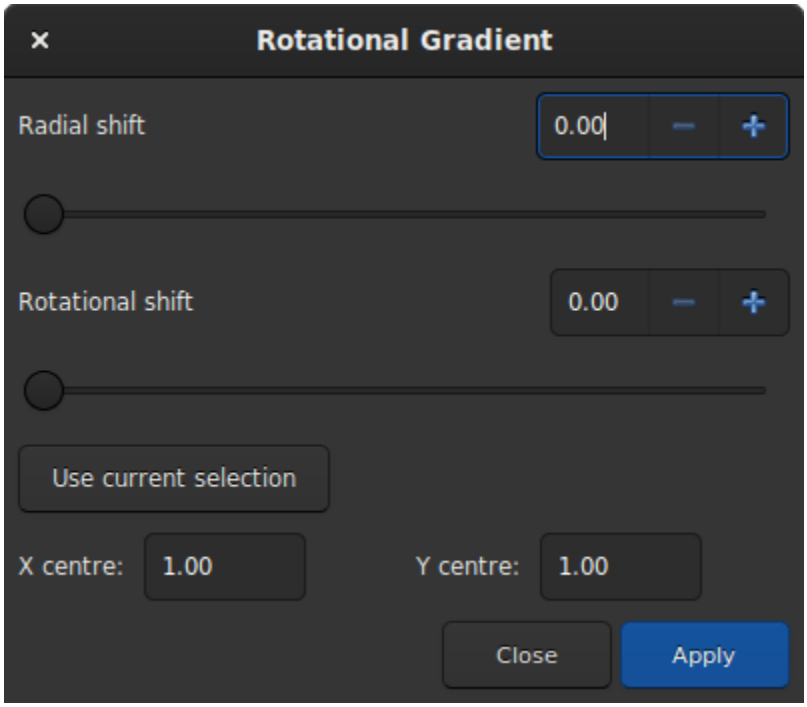


Abb. 62: Dialogbox Larson Sekanina Filter.

Im folgenden Beispiel, das den Kometen C/2022 E3 zeigt, wurde die Ausrichtung auf den Kometen vorgenommen, und die Sterne zeigen deutlische Strichspuren. Der Komet ist sehr kreisförmig und es ist schwierig, Details seiner Aktivität zu erkennen. Daher ist es nicht notwendig, eine radiale Verschiebung zu bestimmen. Für die Drehung wurde ein Winkel von etwas mehr als 15° gewählt (diese Wahl wurde nach mehreren Versuchen und unter Verwendung der Rückgängig-Taste getroffen). Um die Koordinaten des Rotationszentrums zu bestimmen, markieren Sie einfach den Kometenkern und klicken Sie auf *Verwende aktuelle Auswahl*. Diese Aktion kopiert die Koordinaten des Schwerpunkts an die gewünschte Stelle.

Ein einfacher Klick auf *Anwenden* wendet den Filter an. In unserem Beispiel wird ein Jet sichtbar.

Abb. 63: Bild eines Kometen (C/2022 E3 (ZTF)) mit Koma und Schweif (mit freundlicher Genehmigung von Stéphane Garro).

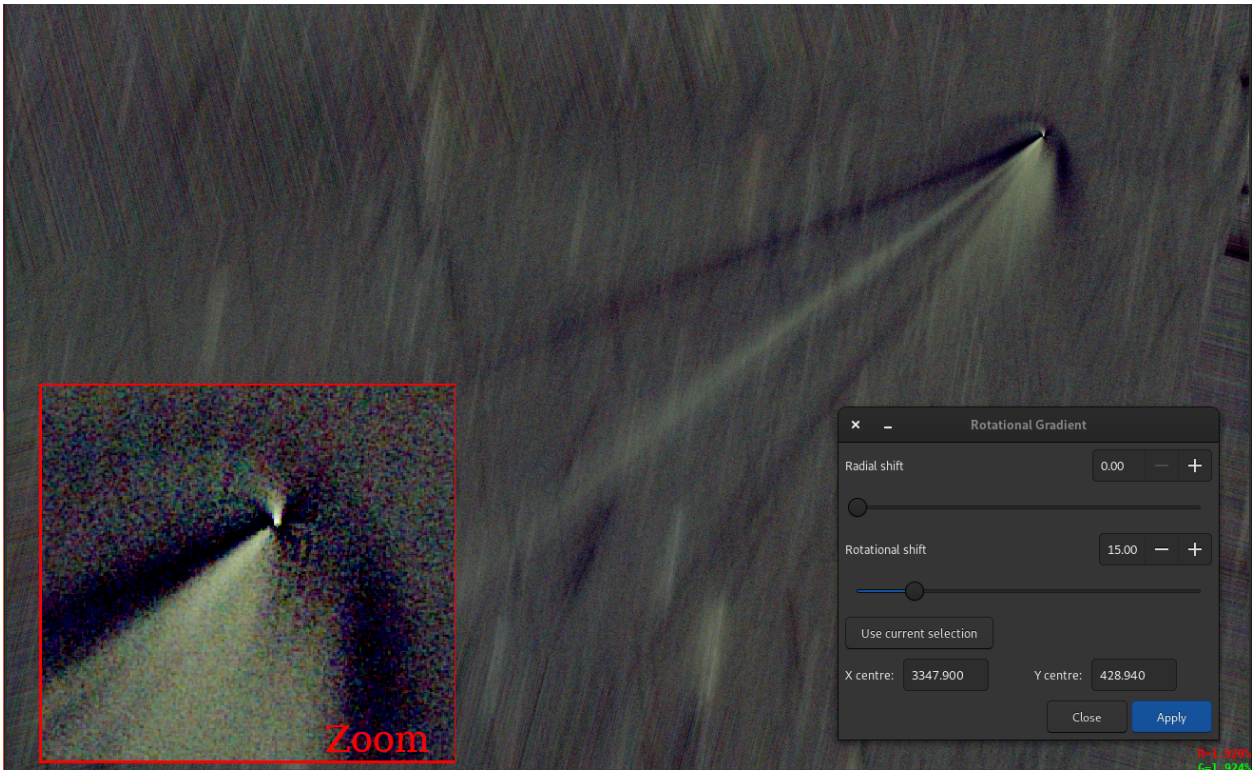


Abb. 64: Nach Anwendung des Filters erscheint der Jet des Kometen sehr deutlich.

Siril Kommandozeile

```
rgradient xc yc dR dalpha
```

Erzeugt zwei Bilder mit einer radialen Verschiebung (**dR** in Pixeln) und einer Rotationsverschiebung (**dalpha** in Grad) in Bezug auf den Punkt (**xc**, **yc**).

Zwischen diesen beiden Bildern haben die Verschiebungen die gleiche Amplitude, aber ein entgegengesetztes Vorzeichen. Die beiden Bilder werden dann addiert, um das endgültige Bild zu erhalten. Dieses Verfahren wird auch Larson-Sekanina-Filter genannt

10.4.11 Magentakorrektur-Filter

This is a cosmetic filter designed to reduce the effect of purple fringing around stars that may be visible when using an achromat refractor especially when broadband imaging. It operates on stretched colour images.

Tipp

Dieser Filter ist für die Bearbeitung nichtlinearer Bilder ausgelegt. Daher muss er unbedingt nach der Streckung des Histogramms angewendet werden.

Der Filter bietet zwei Modi an, die seine Wirkung nur auf Sterne limitieren oder auf alles anwenden.

It reduces purple fringing by applying a luminance mask to the blue channel based on the green channel, and by allowing you to reduce the blue channel intensity. When in starmask mode, it limits the effects to areas around identified stars.

Global

Dieser Modus ist nützlich, wenn man mit einer Sternmaske oder einem Bild arbeitet, das nur Sterne enthält, beispielsweise ein Kugelsternhaufen. Er ist nicht so nützlich für Nebel oder Bilder von Galaxien, in denen die Farbe verändert dadurch werden könnte.

To use, you should reduce the *Blue Adjustment* slider until the effects are optimal. This is generally around 0.15.

You may also increase the *Threshold* slider to restrict the effects to those pixels where the luminance is above this value. In this mode, the optimal value will be quite low.

Sternenmaske

Dieser Modus ist nützlich, um bei kombinierten Bildern den Effekt dieses Filters nur auf Sterne zu begrenzen.

To use, you should tick *Use Starmask*. If you've not already defined a starmask using the Dynamic PSF function, there will be a delay of a few seconds whilst the starmask is generated. You may now temporarily reduce the *Blue Adjustment* to zero. Your stars should now be yellow. If you still see purple fringing, you should increase the Threshold slider until the fringing is entirely covered by yellow. Now you can increase the *Blue Adjustment* slider until your stars are not yellow but your purple fringing is reduced or eliminated, typically around 0.12-0.15.

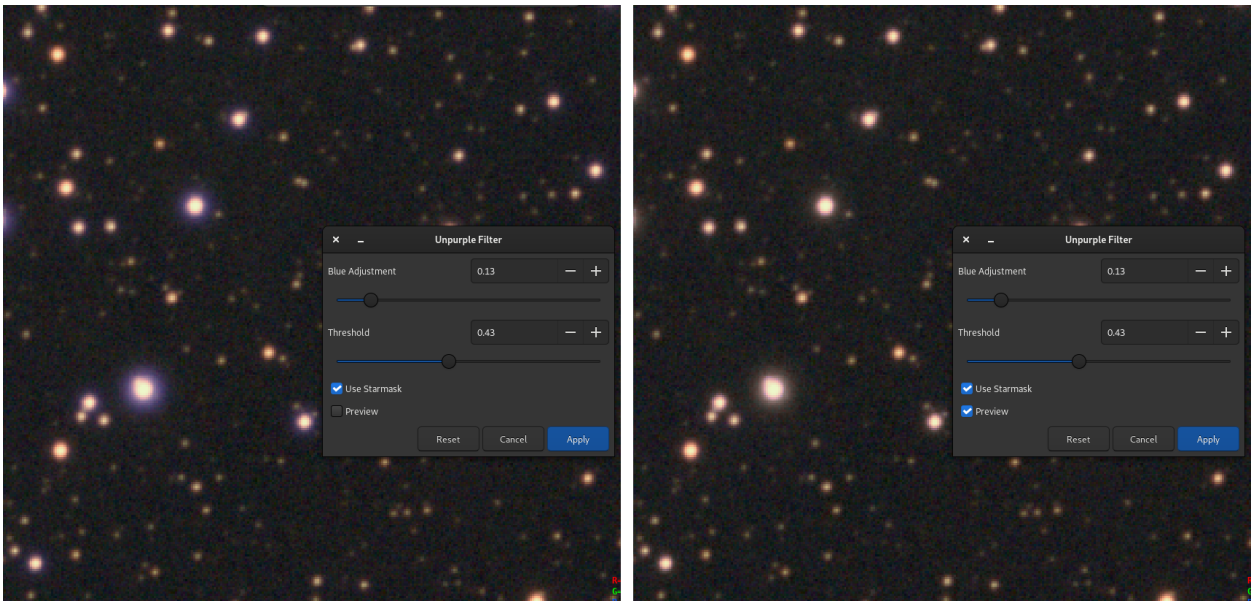


Abb. 65: Comparison before/after application of the unpurple filter. We can see very clearly that the stars have a better color, without the purple ring.

Siril Kommandozeile

```
unpurple [-starmask] [-blue=value] [-thresh=value]
```

Wendet einen kosmetischen Filter an, um die Auswirkungen violetter Farbsäume auf Sternen zu verringern.

Wenn der Parameter **-starmask** angegeben ist, wird eine Sternmaske verwendet, um die zu beeinflussenden Bildbereiche zu identifizieren. Wenn bereits eine dynamische PSF ausgeführt wurde, wird diese für die Sternmaske verwendet, andernfalls wird automatisch eine erstellt. Der Parameter **-mod=** sollte einen Wert um 0,14 erhalten, um die Menge an Lila zu reduzieren. **-thresh=** gibt den Größenmodifikator für jeden Stern in der

Sternmaske an und sollte groß genug sein, damit die Sterne vollständig verarbeitet werden, ohne dass lila Ränder verbleiben. Der Wert sollte zwischen 0 und 1 liegen, normalerweise um 0,5.

Wenn der Parameter **-starmask** nicht angegeben ist, wird die Violettreduzierung auf das gesamte Bild für alle violetten Pixel angewendet, deren Luminanzwert höher ist als der angegebene **-thresh=**. In diesem Fall sollte der **-thresh=**-Wert angemessen niedrig sein. Dieser Modus ist nützlich für Sternmasken oder andere Bilder ohne Nebel oder Galaxien

Verweis: *psf*

10.5 Sternbearbeitung

Sterne sind ein wesentlicher Bestandteil von Deep-Sky-Bildern und spielen eine entscheidende Rolle, wenn es darum geht, die Schönheit und die Details von Himmelsobjekten hervorzuheben. Sie erscheinen oft als brillante Lichtpunkte, die ihre Helligkeit und Farben zur Geltung bringen und Deep-Sky-Bilder wirklich fesselnd machen. Aufgrund der begrenzten Beobachtungsbedingungen können die Sterne auf den Bildern jedoch größer und überbelichtet erscheinen. Um dem entgegenzuwirken, setzen die Astronomen fortschrittliche Bildverarbeitungstechniken ein, um die Sterne separat zu verarbeiten und ihre Größe und Helligkeit im endgültigen Bild zu kontrollieren.

Dieser Teil der Dokumentation ist allem gewidmet, was mit der Verarbeitung der Sterne zu tun hat.

10.5.1 StarNet++ Sternentfernung

Warnung

The Starnet interface built into Siril itself is now deprecated. StarNet 2.5 has been released and is supported by means of a Python script which is available in the scripts repository. The legacy C interface will **not** support Starnet 2.5 but will remain available in Siril 1.4.x for users of older versions of Starnet.

In Siril 1.6 the legacy C interface will be removed so all users will need to upgrade to Starnet 2.5 and use the python script. The script does provide CLI args so where scripts use the Siril `starnet` command they will need to adapt to using `pyscript StarNet.py` instead.

StarNet++ ist eine Software, die von Nikita Misiura entwickelt wurde. Die erste Version <https://github.com/nekitmm/starnet>`_ wurde unter einer freien und Open-Source-Lizenz veröffentlicht. Leider wurde die Version 2 proprietär und die Quellen sind seitdem geschlossen. Die Version 2 ist kostenlos erhältlich unter <https://www.starnetastro.com/download>`_. Stellen Sie sicher, dass Sie die **Kommandozeilenversion** herunterladen. Siril kann mit jeder Version des StarNet CLI-Tools zusammenarbeiten, einschließlich der neuen experimentellen Torch-basierten Version, die zunächst für M1- und M2-basierte Apple Macs veröffentlicht wurde.

Warnung

Wenn Sie sich fragen, **warum StarNet nicht startet**, führen Sie es bitte zuerst außerhalb von Siril aus. Es ist nicht Sirils Schuld, wenn es von Ihrem Computer nicht unterstützt wird oder aus irgendeinem Grund falsch installiert ist. Wenn Ihr Prozessor die von StarNet benötigten Vektorisierungsanweisungen nicht unterstützt, gibt es keine Möglichkeit, dies zu umgehen. Die Fehlermeldung wird angezeigt, wenn Sie StarNet direkt ausführen.

Tipp

Damit Siril unter **MacOS** StarNet korrekt erkennen und verwenden kann, müssen zunächst einige Berechtigungs- und Sicherheitsprobleme behoben werden. Öffnen Sie zunächst die Anwendung „Terminal“ im Ordner „Dienstprogramme“ unter „Programme“. In Terminal müssen Sie Ihr Arbeitsverzeichnis von Ihrem Home-Verzeichnis in das StarNetCLI-Installationsverzeichnis ändern. Geben Sie dazu `cd` gefolgt von einem `space` ein und ziehen Sie dann den StarNetCLI-Ordner in das Terminal-Fenster, um seinen Pfad zu kopieren. Drücken Sie `enter`. Geben Sie dann die folgenden vier Befehle ein und drücken Sie nach jedem Befehl `enter`:

```
xattr -r -d com.apple.quarantine libtensorflow_framework.2.dylib
xattr -r -d com.apple.quarantine starnet++
chmod +x starnet++
chmod +x run_starnet.sh
```

Dann kann bei der ersten Verwendung mit Siril die Ausführung von StarNet++ mit einer Warnung über libtensorflow fehlschlagen. Schließen Sie diese Warnung. Öffnen Sie die Systemeinstellungen und klicken Sie unter Datenschutz und Sicherheit auf die Schaltfläche *Trotzdem erlauben* für libtensorflow. Danach sollte StarNet++ in Siril ordnungsgemäß ausgeführt werden.

Tipp

Für **MacOS** gibt es wieder eine neue Starnet-Datei, die für den Apple Silicon Chip optimiert ist und auf der Website veröffentlicht wurde: https://www.starnetastro.com/experimental/. Diese neue Version ist viel schneller als die vorherige Version, da sie das neue MPS-beschleunigte PyTorch (https://developer.apple.com/metal/pytorch/) verwendet. Außerdem enthält diese Version signierte Binärdateien, folgen Sie den Installationsanweisungen in der README.txt

Es ist jedoch immer noch möglich, dass Siril externe Binärdateien ausführt, und wir haben beschlossen, dies ab Siril 1.2.0 zu implementieren. Für die Einstellungen lesen Sie bitte die *Seite* Dort wird erklärt, wie man Siril mitteilt, wo sich StarNet befindet.

Warnung

Dies ist das Verzeichnis der Kommandozeilenversion von Starnet++, das angegeben werden muss, nicht das der GUI-Version.

Beachten Sie, dass StarNet seine Eingabe in Form von TIFF-Bildern benötigt. Wenn Siril also ohne libtiff-Unterstützung kompiliert wird, ist die StarNet-Integration nicht verfügbar.

Der Hauptzweck von StarNet besteht darin, alle Sterne aus den Bildern zu entfernen, um einen unterschiedliche Prozess auf die Sterne und die übrigen Bildinhalte anzuwenden. Dies hilft in der Regel, die Aufblähung der Sterne während der verschiedenen Abschnitte zu kontrollieren, ist aber auch sehr nützlich für die Erstellung von Bildern von Kometen, bei denen sich die Kometenverfolgungsrate erheblich von den weit entfernten Sternen unterscheiden kann.

Das Tool ist sehr einfach zu bedienen, und es sind nur fünf Optionen verfügbar:

- **Lineares Bild vordehnen**: Wenn diese Option ausgewählt ist, wird vor der Ausführung von StarNet++ eine optimierte MTF-Streckung (Midtone Transfer Function) auf das Bild angewendet, und nach Abschluss wird die umgekehrte Streckung durchgeführt. Dies ist für die Verwendung von StarNet während der linearen Verarbeitungsphase erforderlich.
- **Sterne nach Fertigstellung neu zusammensetzen**: Wenn diese Option ausgewählt ist, wird nach Abschluss der Sternentfernung das Werkzeug zur Neuzusammensetzung der Sterne geöffnet, das eine Schnittstelle zum unabhängigen Strecken und Überblenden des Hintergrunds und der Sterne bietet, wenn eine Reduzierung der Sterne statt einer vollständigen Entfernung gewünscht ist. Diese Option hat bei der Bearbeitung einer Sequenz keine Auswirkungen.

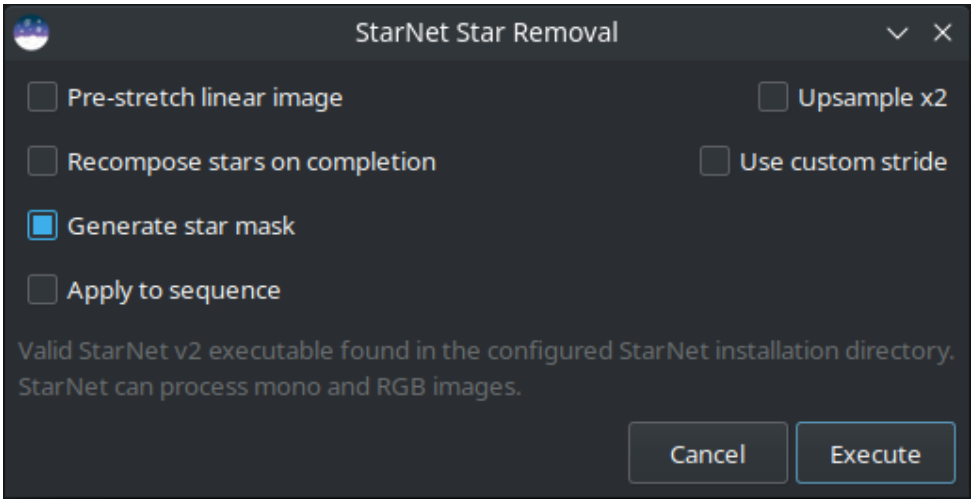


Abb. 66: Dialogbox StarNet.

- **Generiere Sternmaske**: Dies erzeugt eine Sternmaske und speichert sie im Arbeitsverzeichnis. Die Sternmaske wird als Differenz zwischen dem Originalbild und dem sternlosen Bild berechnet. Standardmäßig wird eine Sternmaske erzeugt.
- **Resample x2**: Mit dieser Option wird das Bild vor der Ausführung von StarNet++ um den Faktor 2 hochgerechnet. Dies verbessert die Leistung bei sehr engen stehenden Sternen, vervierfacht aber die Verarbeitungszeit und kann die Leistung bei sehr großen Sternen beeinträchtigen. Das Bild wird nach der Bearbeitung wieder auf die ursprüngliche Größe skaliert.
- **Benutzerdefinierte Schritte verwenden**: Für den **stride**-Parameter in StarNet++ kann ein benutzerdefinierter Wert eingegeben werden. Der Standardwert ist 256 und der StarNet-Entwickler empfiehlt, diesen Wert nicht zu ändern.

Der StarNet-Prozess kann leicht auf eine Sequenz angewendet werden. Mit der Checkbox *Auf Sequenz anwenden* wird ausgewählt, ob der Prozess auf ein einzelnes Bild oder auf eine Sequenz angewendet werden soll. Wenn der Prozess auf eine Sequenz angewendet wird, wird eine neue Sequenz mit den sternlosen Bildern erstellt, und wenn die Sternmaskenerzeugung ausgewählt wird, wird eine zweite Sequenz mit den entsprechenden Sternmaskenbildern erstellt.

Weitere Informationen über StarNet finden Sie auf der Original-Webseite.

Mit einem Klick auf *Ausführen* wird der Prozess gestartet. Je nach der Leistung Ihres Rechners kann dies langsam sein. Siril zeigt jedoch einen Fortschrittsbalken an, um die Verarbeitung zu verfolgen. Wie bei anderen Siril-Prozessen wird der Fortschrittsbalken bei der Verarbeitung einer Sequenz erst nach Abschluss jedes Bildes der Sequenz aktualisiert und zeigt den Gesamtfortschritt der Sequenz an.

Befehle

Siril Kommandozeile

```
starnet [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Dieser Befehl ruft StarNet auf, um Sterne aus dem aktuellen Bild zu entfernen.

Voraussetzung: StarNet ist ein externes Programm, das nicht automatisch mit Siril installiert wird, und muss vor der ersten Verwendung dieses Befehls korrekt installiert werden, wobei der Pfad zu seinem

Installationsverzeichnis in Einstellungen / Verschiedenes korrekt eingestellt sein muss. Das Verzeichnis muss die Kommandozeilenversion des Programms enthalten (nicht die GUI-Version, die für Windows-Benutzer existiert).

Das sternlose Bild wird bei Fertigstellung geladen, und ein Bild mit der Sternmaske wird im Arbeitsverzeichnis erstellt, sofern nicht der optionale Parameter **-nostarmask** angegeben wird.

Optional können dem Befehl auch Parameter übergeben werden:
- The option **-stretch** is for use with linear images and will apply a pre-stretch before running StarNet and the inverse stretch to the generated starless and starmask images.
- To improve star removal on images with very tight stars, the parameter **-upscale** may be provided. This will upsample the image by a factor of 2 prior to StarNet processing and rescale it to the original size afterwards, at the expense of more processing time.
- The optional parameter **-stride=value** may be provided, however the author of StarNet *strongly* recommends that the default stride of 256 be used

Siril Kommandozeile

```
seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Dieser Befehl ruft Starnet++ auf, um Sterne aus der Sequenz **sequencename** zu entfernen. Siehe STARNET

Verweis: *starnet*

10.5.2 Stern-Neuzusammensetzung

Stern-Neuzusammensetzung ist ein GUI-Werkzeug, das bei der Kombination von Bildern ohne Sterne und Sternmasken hilft. Es bietet keine einzigartige Bildmanipulation, die nicht auch auf andere Weise möglich ist, z. B. mit PixelMath und dem Werkzeug "Verallgemeinerte hyperbolische Streckung", aber es bietet eine Echtzeit-Vorschau auf die Kombination von zwei separaten Bildern mit unterschiedlichen Streckungen, die auf beide angewendet werden.

Es gibt kein Kommandozeilen-Äquivalent für dieses Werkzeug, da es rein grafisch ist. Allerdings können sternlose und Sternmaskenbilder mit den Befehlen *pm* und GHT-bezogenen Befehlen (*ght*, *invght*, *modasinh*, *invmodasinh* und *linstretch*) kombiniert werden.

Das Werkzeug befindet sich im Menü Bildbearbeitung im Untermenü Sternbearbeitung.

Das Dialogfeld ist in zwei Spalten unterteilt, eine für jedes der Eingabebilder.

Jedes Eingabebild wird über die entsprechende Dateiauswahl geladen. Jede Spalte verfügt über eine Vorschau des Streckungshistogramms, die minimiert werden kann, um die Verwendung auf kleinen Bildschirmen zu erleichtern, sowie über eine Reihe von GHS-Streckungssteuerungen sowie Schaltflächen zum Zurücksetzen und Anwenden.

Der Histogrammmodus kann zwischen linear und logarithmisch umgeschaltet werden, indem Sie den Schalter am unteren Rand des Dialogs betätigen. Dieses Dialogfeld befolgt die Siril-weite Voreinstellung für lineare oder logarithmische Histogramme, die im Fenster "Voreinstellungen" festgelegt werden kann.

Tipp

Star Recomposition does not obey the FITS 16-bit preference as there could be an adverse effect on performance in some cases (the calculations must be done using floating point maths anyway). So if you really want a 16-bit result

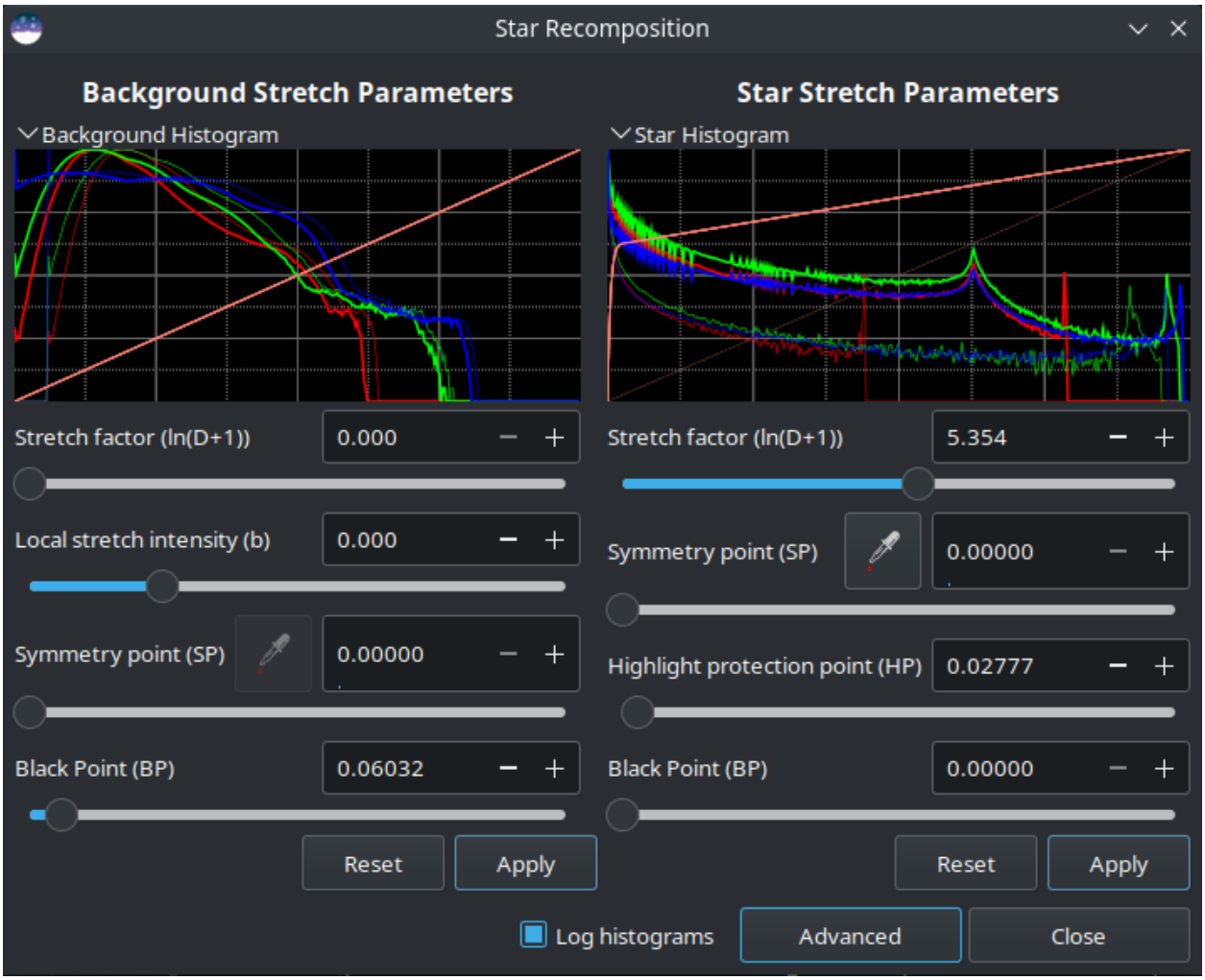


Abb. 67: Stern-Neuzusammensetzung.

to be created from 32-bit input images then you will need to use the *precision switch*.

Einfacher Modus

Das Dialogfeld hat zwei Ansichten, die bestimmen, welche Steuerelemente angezeigt werden. Es öffnet sich im einfachen Modus, der nur die nützlichsten Steuerelemente für eine typische Starless/Starmask-Kombination anzeigt.

- Der Streckungstyp für das sternlose Bild ist auf verallgemeinerte hyperbolische Streckung eingestellt und die Steuerelemente Streckungsfaktor, lokale Streckungsintensität, Symmetriepunkt und Schwarzpunkt werden angezeigt. Neben der SP-Steuerung kann der Symmetriepunkt auch mit der Pipette eingestellt werden, um den durchschnittlichen Pixelwert einer Auswahl aus dem Bild auszuwählen.

  Beachten Sie, dass die Pipette bei einer nicht angewendeten BP-Verschiebung deaktiviert ist: Aufgrund des Prozesses der Anwendung der hyperbolischen Streckung und der anschließenden BP-Verschiebung wird das Verhalten des Werkzeugs nicht intuitiv, wenn ein BP-Parameter ungleich Null eingestellt ist. Um dies zu beheben, wenden Sie einfach die BP-Verschiebung an, und die Pipette wird für die nächste hyperbolische Streckung wieder verfügbar.

- Der Streckungstyp für das Sternmaskenbild ist auf modifizierte Arcsinh-Streckung eingestellt und die Steuerelemente Streckungsfaktor und Lichter-Schutzpunkt werden angezeigt.
- Das menschlich gewichtete Luminanz-Farbmodell wird für beide Streckungssätze verwendet: Dadurch bleiben die Farben im ungestreckten Bild besser erhalten.

Einzelheiten zu allen Streckungssteuerungen, sowohl die im einfachen als auch die im erweiterten Modus, finden Sie auf der Dokumentationsseite Generalisierte hyperbolische Streckung.

Das BP-Steuerelement funktioniert etwas anders als das BP-Steuerelement in der eigenständigen linearen GHS-Streckung (Schwarzpunktanpassung). In diesem Werkzeug wird die Schwarzpunktanpassung *nach* der hyperbolischen Streckung angewendet, während sie im eigenständigen Werkzeug eine separate Streckung ist, die allein angewendet wird. Bei dem Versuch, die Kombination unabhängiger Streckungen für die beiden Eingabebilder zu optimieren, erwies sich diese Vorgehensweise als die praktikabelste. Dies bedeutet jedoch, dass der Betrag der Schwarzpunktverschiebung, der in diesem Werkzeug erforderlich ist, sich von dem Betrag unterscheidet, der im GHS-Werkzeug erforderlich ist, und dass der Schwarzpunkt nicht durch Klicken auf das Histogramm eingestellt werden kann.

Jede Streckung ist unabhängig. Die Streckungseinstellungen für das sternlose Bild können über die linke Schaltfläche "Anwenden" angewendet werden: Dadurch wird das sternlose Bild entsprechend den aktuellen Streckungseinstellungen gestreckt und dann die Streckungseinstellungen zurückgesetzt, so dass weitere Streckungen iterativ angewendet werden können. In ähnlicher Weise können die Streckungseinstellungen für die Sternmaske über die rechte Schaltfläche "Anwenden" angewendet werden. Beide Streckungseinstellungen können mit der jeweiligen Schaltfläche "Zurücksetzen" auf die Standardwerte zurückgesetzt werden.

Mit der Schaltfläche am unteren Rand des Dialogs kann zwischen dem einfachen und dem erweiterten Modus umgeschaltet werden.

Erweiterte Ansicht

Im erweiterten Modus steht die gesamte Palette der GHS-Stretch-Steuerungen zur Verfügung, einschließlich Stretch-Typ, Farb-Stretch-Modell und Schattenschutzpunkt für beide Eingangsbilder. Dies ermöglicht bei Bedarf eine stärkere Anpassung der beiden Streckungen. Wenn die Benutzeroberfläche in den einfachen Modus zurückgesetzt wird, bleiben alle Änderungen, die mit den erweiterten Steuerelementen vorgenommen wurden, in Kraft, nur die Steuerelemente werden ausgeblendet.

Bemerkung

Es ist nicht möglich, die Sättigungskanäle mit diesem Werkzeug zu strecken. Das Werkzeug ist bereits recht speicher- und rechenintensiv: Eine Verdoppelung des Speicherbedarfs durch Hinzufügen einer HSL-Kopie jedes Arbeitsbildes wird als übertrieben angesehen. Die Sättigung kann leicht separat gestreckt werden, nachdem die Kombination abgeschlossen ist.

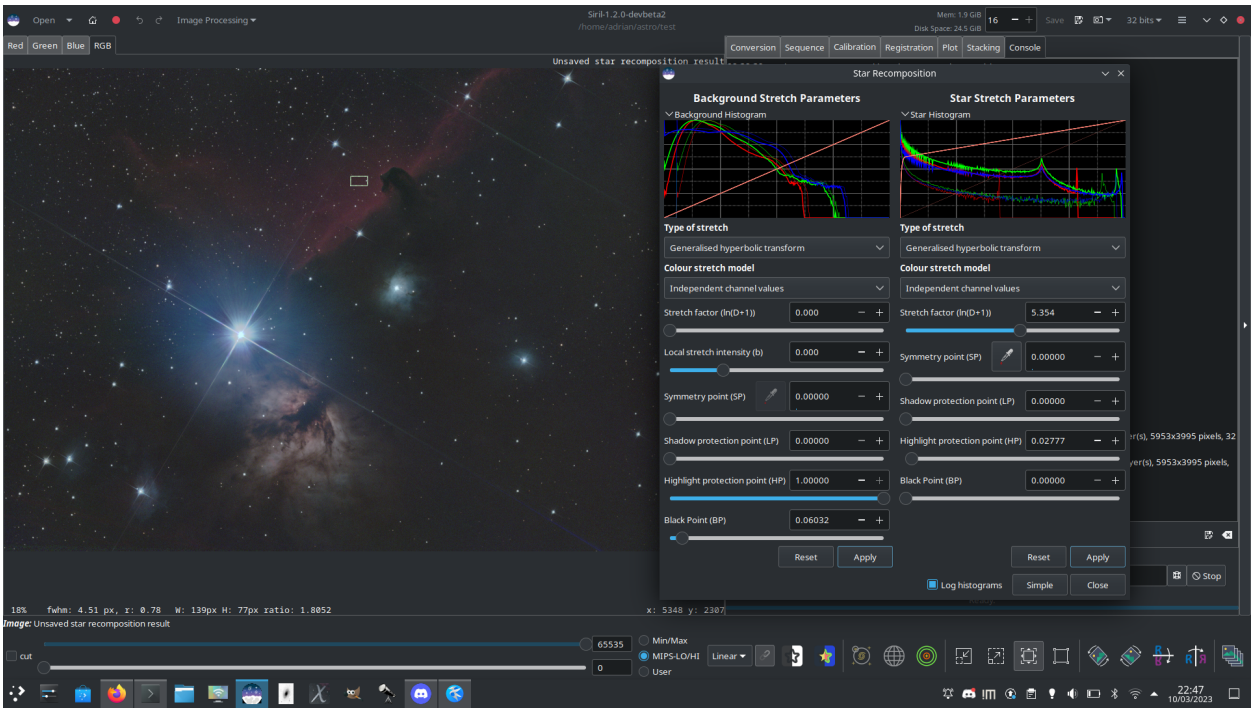


Abb. 68: Mit der Sternneuzusammensetzung ein sternloses Bild und eine Sternmaske der Alnitak-Region kombinieren

10.5.3 Sterne entsättigen

Wenn ein Sternsucher auf ein Bild angewendet wird (dessen Daten immer linear sind), werden um die Sterne Ellipsen angezeigt. Wenn eine Ellipse magentafarben ist, bedeutet dies, dass der Stern gesättigt ist.

Ein gesättigter Stern ist ein Stern, dessen hellste Pixel keine Informationen mehr über die Farbe haben und auf den Maximalwert beschnitten werden. Im Allgemeinen versuchen wir, die Sterne nicht zu sättigen, auch wenn dies bei den hellsten Sternen nicht immer möglich ist. Wenn trotz aller Vorsichtsmaßnahmen immer noch gesättigte Sterne vorhanden sind, verfügt Siril über einen Algorithmus, der das Profil des Sterns unter Berücksichtigung der Ergebnisse der während der Sternsuche ermittelten Daten rekonstruiert.

Zunächst müssen Sie eine Sternsuche durchführen, entweder mit dem Befehl *findstar* oder mit der Schaltfläche im Fenster *Dynamisches PSF*. Das Entsättigungswerkzeug finden Sie dann unter *Sternverarbeitung* → *Sterne entsättigen*.

Tipp

Wir empfehlen die Verwendung eines Moffat-Profils im Fenster *Dynamisches PSF*, um bessere Parameter zu erhalten.

Warnung

Es ist wichtig, dieses Tool auf linearen Bildern auszuführen, da die Sterne sonst kein Gauß/Moffat-Profil haben und die Berechnungen ungültig sind.

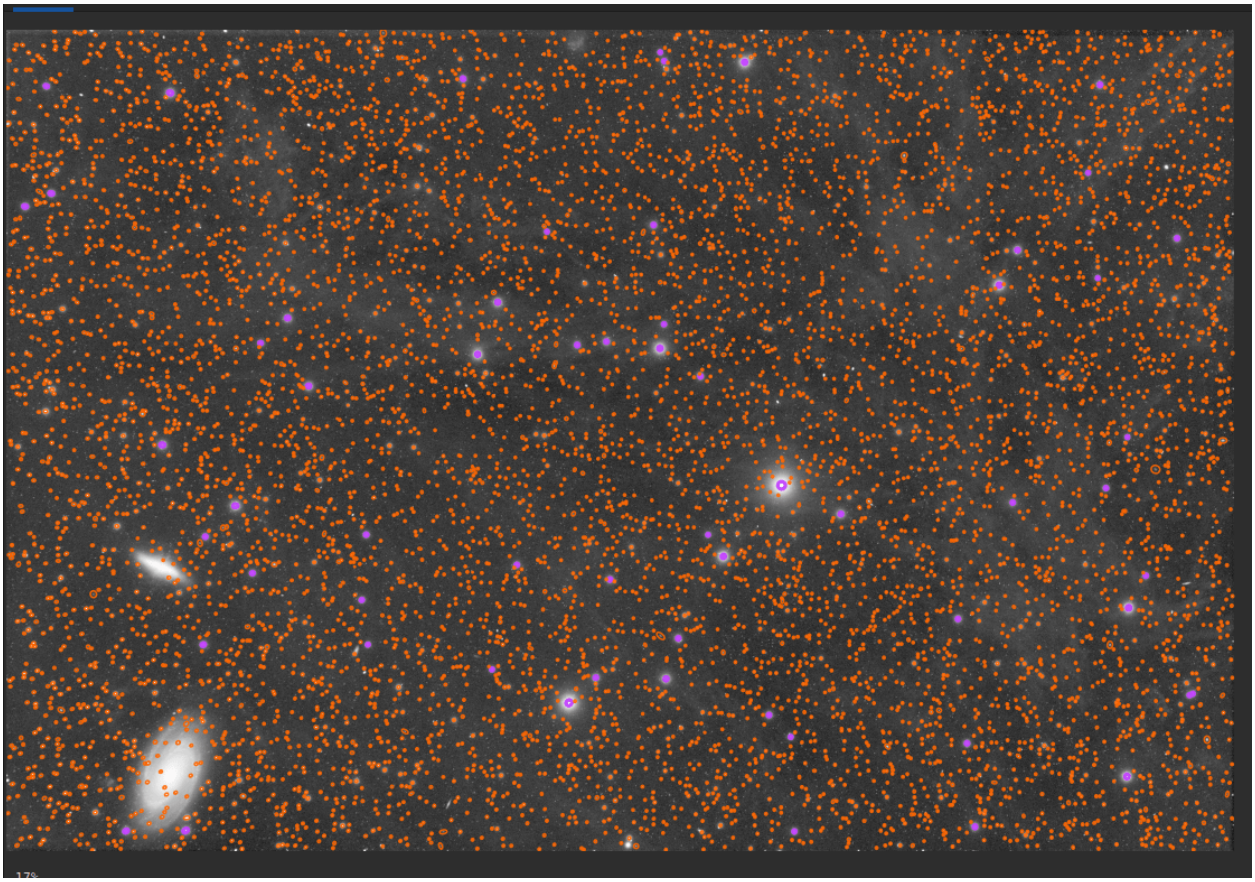


Abb. 69: Eine Sternsuche zeigt alle von Siril gefundenen Sterne. Die magentafarbenen Ellipsen stehen für gesättigte Sterne. Das Bild wird in der Autostretch-Ansicht angezeigt: Die Daten sind noch linear.

Nach einem Klick auf das Werkzeug schaltet Siril auf die Konsole um und zeigt die Ergebnisse des aktuellen Prozesses an:

```
22:26:17: Star synthesis (desaturating clipped star profiles): processing...
22:26:17: Findstar: processing for channel 0...
22:26:21: Star synthesis: desaturating stars in channel 0...
22:26:21: Star synthesis: 70 stars desaturated
22:26:21: Remapping output to floating point range 0.0 to 1.0
22:26:21: Execution time: 4.09 s
```

Um die Änderungen zu sehen, ist es notwendig, eine erneute Sternsuche durchzuführen.

Siril Kommandozeile

```
unclipstars
```

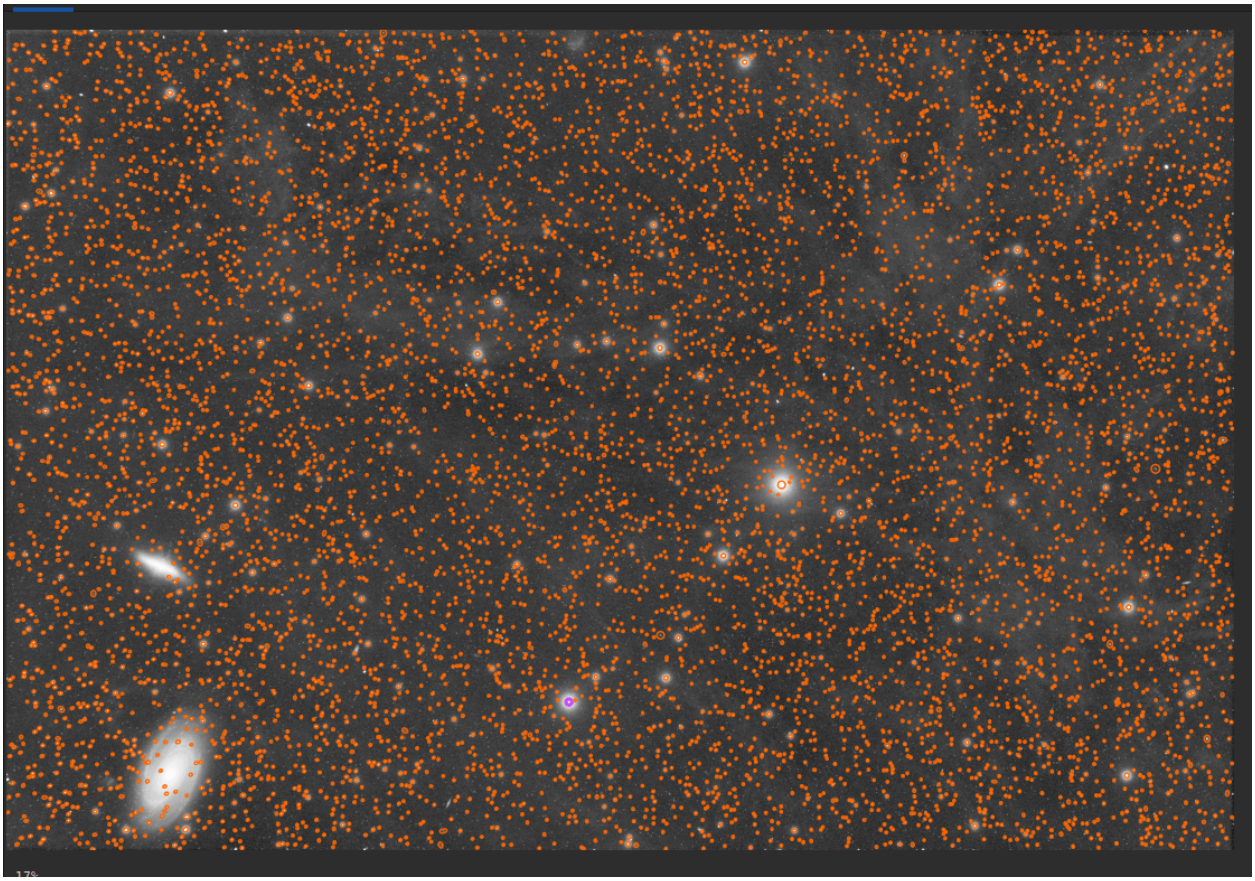


Abb. 70: Nach einer Entsättigungsbearbeitung sind keine magentafarbenen Ellipsen mehr sichtbar. Alle Sterne sind rekonstruiert worden. Das Bild wird in der Autostretch-Ansicht angezeigt: Die Daten sind immer noch linear.

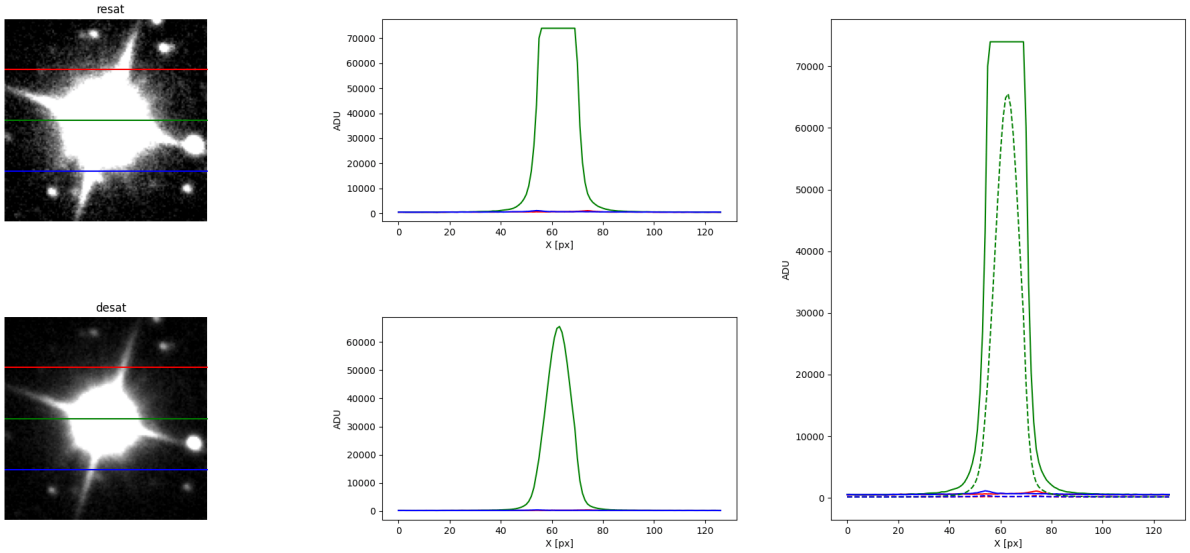


Abb. 71: Vergleich für einen Stern vor und nach der Anwendung des Entsättigungswerkzeugs.

Reprofiliert übersättigte Sterne, um sie zu entsättigen, und skaliert die Ausgabe so, dass alle Pixelwerte <= 1,0 sind

10.5.4 Vollständige Resynthese

Das Werkzeug Vollständige Resynthese soll helfen, stark verzerrte Sterne mit Hilfe der Sternanpassungsfunktionen von Siril zu korrigieren. Es kann bei der Rettung von Bildern hilfreich sein, die unter einer starken Koma oder anderen Verzerrungen leiden. Wenn Siril die Sterne erkennen kann, kann es sie korrigieren.

Das Werkzeug befindet sich im Menü Bildbearbeitung im Untermenü Sternbearbeitung.

Das Ergebnis des Werkzeugs ist eine synthetische Sternmaske. Um diese nutzen zu können, muss sie mit einer sternfreien Version des Originalbildes neu kombiniert werden. Diese kann mit dem Befehl *starnet* oder dem Starnet-GUI-Tool oder mit Sternentfernungssoftware von Drittanbietern erstellt werden.

Dieses Werkzeug hat keine Optionen, man klickt einfach auf den Menüpunkt, um es zu benutzen, oder benutzt den Befehl *synthstar*.

Wenn keine Sterne im Bild erkannt wurden, ermittelt das Programm automatisch Sterne unter Verwendung der aktuellen Sternmodellierungsparameter, die über das Werkzeug "Dynamische PSF" oder über den Befehl *setfindstar* zugänglich sind.

Wenn Sterne mit dem Werkzeug Dynamische PSF oder dem Befehl *findstar* modelliert wurden, werden die entdeckten Sterne mit ihren individuellen modellierten Leuchtkraftprofilen resynthetisiert. Eine Verknüpfung zum Werkzeug "Dynamische PSF" wird über die Konfigurationsschaltfläche im GUI-Menü neben dem Werkzeug "Vollständige Resynthese" bereitgestellt.

Es wird empfohlen, die Sternerkennung zunächst manuell durchzuführen, da dies eine Überprüfung der Ergebnisse ermöglicht: Falls Galaxien fälschlicherweise als Sterne erkannt wurden, können sie aus der Liste der Sterne entfernt werden, bevor die Resynthese durchgeführt wird.

Sobald die synthetische Sternmaske erstellt wurde, kann sie mit dem Werkzeug Stern-Neuzusammensetzung mit dem sternlosen Bild kombiniert werden.

Befehle

Siril Kommandozeile

```
synthstar
```

Korrigiert schlechte Sterne im geladenen Bild. Unabhängig davon, wie viel Koma, Nachführdrift oder andere Verzerrungen Ihre Sterne haben, wenn Sirils Sternsucherroutine sie erkennen kann, wird synthstar sie korrigieren. Wenn Sie besonders vorsichtig sein wollen, sollten Sie alle Sterne, die Sie korrigieren möchten, manuell auswählen. Dazu können Sie den Konsolenbefehl findstar oder den Dialog Dynamischer PSF verwenden. Wenn Sie die Sternsuche nicht ausgeführt haben, wird sie automatisch mit den Standardeinstellungen ausgeführt.

Die besten Ergebnisse erzielen Sie, wenn Sie Synthstar vor dem Strecken ausführen.

Das Ergebnis von synthstar ist eine vollständig korrigierte synthetische Sternmaske mit perfekt runden Stern-PSFs (Moffat- oder Gauß-Profile je nach Sternsättigung), die so berechnet werden, dass sie mit der Intensität, der FWHM, dem Farbton und der Sättigung übereinstimmen, die für jeden im Eingabebild erkannten Stern gemessen wurden. Diese kann dann mit dem sternlosen Bild rekombiniert werden, um ein Bild mit perfekten Sternen zu erzeugen.

Für diesen Befehl sind keine Parameter erforderlich

Verweis: *psf*

10.6 Geometrie

10.6.1 Rotieren

Um 90 Grad drehen

Mit dem entsprechenden Menü können Sie das Bild um 90 Grad im und gegen den Uhrzeigersinn drehen. Hier wird die Drehung ohne Interpolation der Pixel durchgeführt und ist daher die bevorzugte Methode, wenn Sie das Bild um ein Vielfaches von 90 Grad drehen möchten. Diese Funktion ist auch über die Icons und in der Symbolleiste zu erreichen.

Rotieren und Zuschneiden

Für eine Drehung um einen anderen Winkel müssen Sie das Werkzeug Drehen&Zuschneiden verwenden. Es ermöglicht ein präzises Drehen und Zuschneiden, das leicht kontrolliert werden kann.

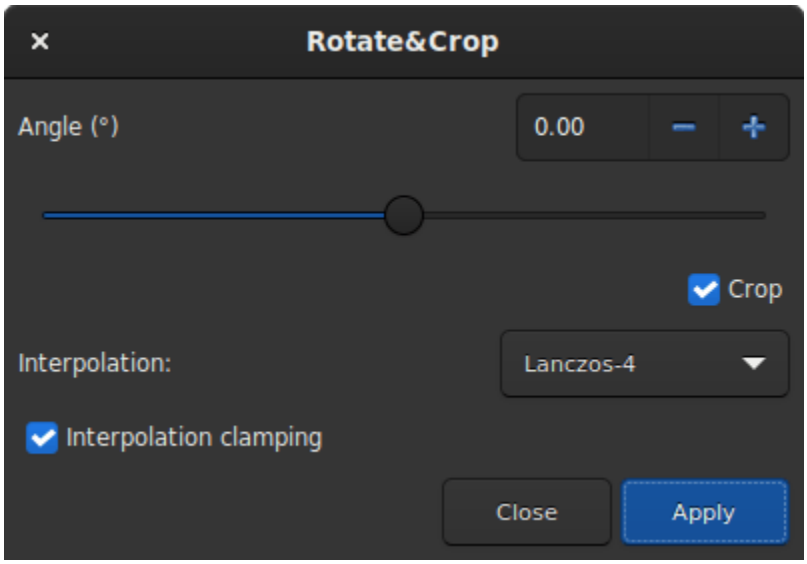


Abb. 72: Dialogbox Drehen und zuschneiden.

Es stehen fünf Interpolationsalgorithmen zur Verfügung:

- Nächster Nachbar
- Bilinear
- Bikubisch
- Pixel zu Fläche Verhältnis
- Lanczos-4 (Standard)

Lanczos-4 ist das Verfahren, das die besten Ergebnisse liefert. Wenn Sie jedoch Artefakte sehen, insbesondere Sterne, die von schwarzen Pixeln umgeben sind, sollten Sie andere Interpolationsverfahren ausprobieren. Die Schaltfläche

Interpolation wendet jedoch einen Begrenzungsfaktor auf die bikubische und Lanczos-4-Interpolation an, um Ringing-Artefakte zu vermeiden.

Wenn Sie nicht möchten, dass das Bild nach der Drehung beschnitten wird, sollten Sie das Häkchen bei der Schaltfläche *Zuschnitt* entfernen. Die fehlenden Bereiche des Bildes werden dann jedoch mit schwarzen Pixeln gefüllt.

Das Interessante an diesem Werkzeug ist, dass die Drehung des Bildes durch einen roten Rahmen dargestellt wird, wie in der Abbildung unten zu sehen ist. Wenn eine Auswahl aktiv ist, ist es außerdem möglich, ihre Größe zu ändern und die Veränderung des Rahmens in Echtzeit zu sehen.

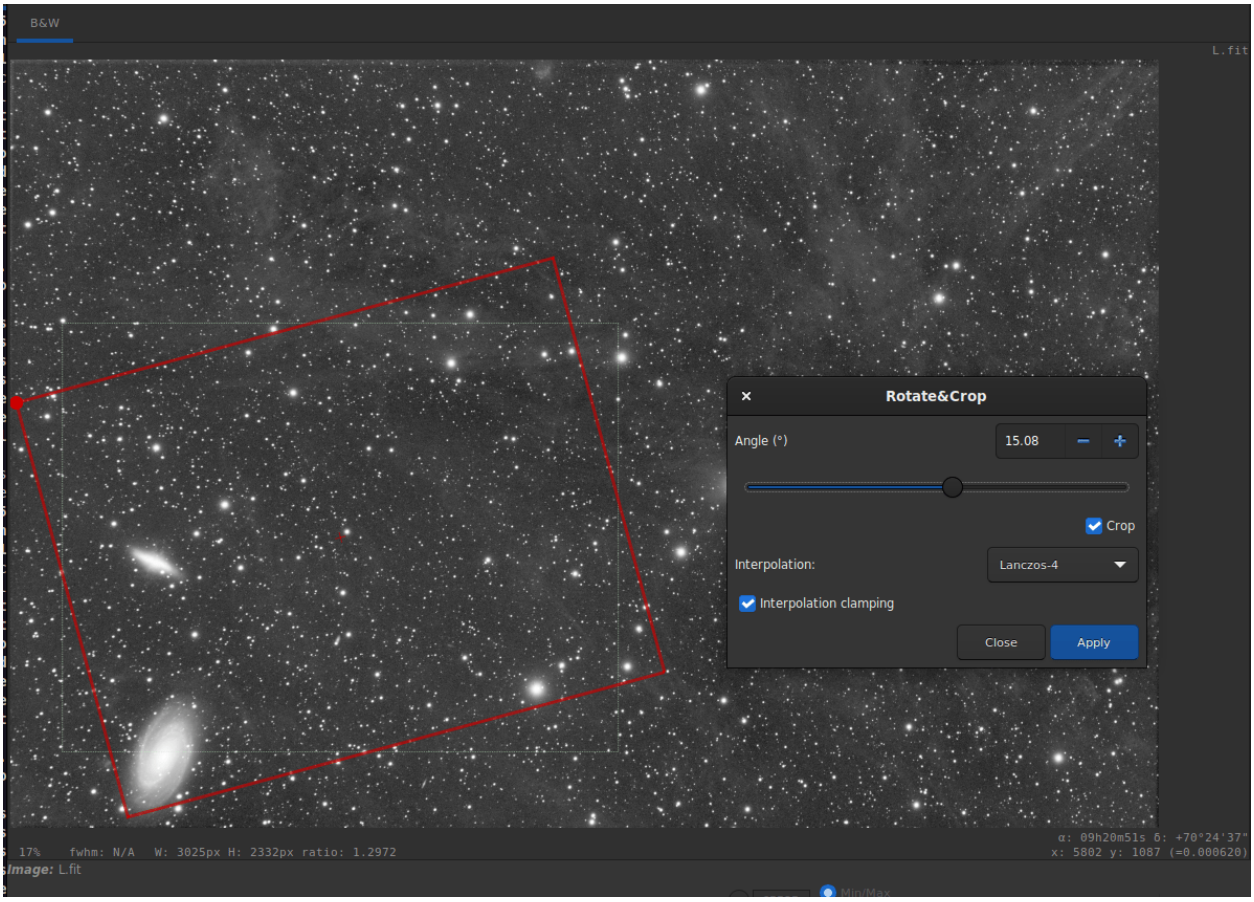


Abb. 73: Dialogfeld "Drehen und Zuschneiden" mit einer aktiven Auswahl. Klicken Sie, um die Abbildung zu vergrößern und die Details besser zu sehen.

Siril Kommandozeile

```
rotatePi
```

Dreht das geladene Bild um 180° um dessen Mitte. Dies ist gleichwertig mit dem Befehl "ROTATE 180" oder "ROTATE -180"

Verweis: *rotate*

Siril Kommandozeile

```
rotate degree [-nocrop] [-interp=] [-noclamp]
```

Dreht das Bild um einen Winkel mit dem Wert **degree**. Die Option **-nocrop** kann hinzugefügt werden, um ein Beschneiden auf die Bildgröße zu vermeiden (es werden schwarze Ränder hinzugefügt).

Hinweis: Wenn eine Auswahl aktiv ist, d. h. durch Verwendung eines Befehls `boxselect` vor `rotate`, wird das resultierende Bild ein gedrehter Ausschnitt sein. In diesem besonderen Fall wird die Option **-nocrop** ignoriert, wenn sie übergeben wird.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet. Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden

10.6.2 Spiegeln

Es ist auch möglich, eine Spiegeltransformation auf das Bild anzuwenden. Entweder entlang der x-Achse oder entlang der y-Achse. Diese Transformation ist auch über die Schaltflächen und der Symbolleiste zugänglich.

Siril Kommandozeile

```
mirrorx [-bottomup]
```

Spiegelt das Bild um die horizontale Achse. Die Option **-bottomup** spiegelt das Bild nur, wenn es nicht bereits "von-unten-nach-oben" ist

Siril Kommandozeile

```
mirrory
```

Spiegelt das Bild um die vertikale Achse

10.6.3 Binning

Das Binning ist eine spezielle Transformation für das Resampling von Bildern. Es berechnet die Summe oder den Mittelwert der Pixel 2x2, 3x3, ... (abhängig vom Binning-Faktor) des gespeicherten Bildes (ähnlich dem analogen Binning einer CCD-Kamera).

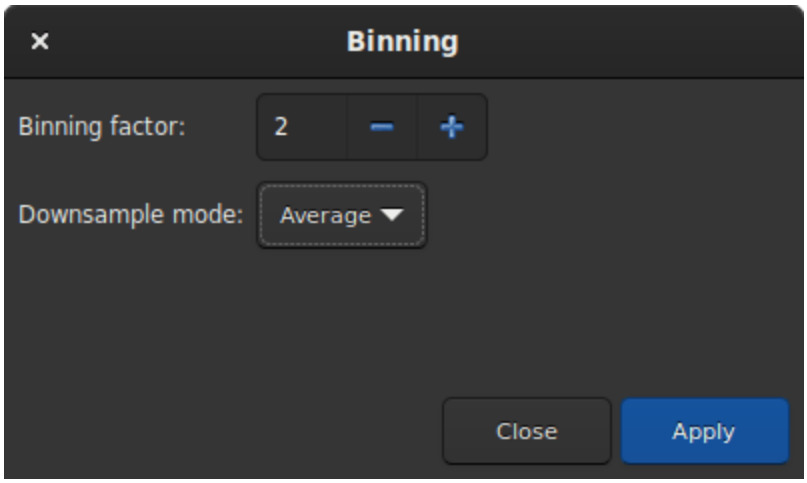


Abb. 74: Dialogbox Binning

Siril Kommandozeile

```
binxy coefficient [-sum]
```

Berechnet das numerische Binning des geladenen Bildes (Summe der Pixel 2x2, 3x3..., ähnlich dem analogen Binning einer CCD-Kamera). Wenn das optionale Argument **-sum** übergeben wird, wird die Summe der Pixel berechnet, ansonsten der Durchschnitt der Pixel

10.6.4 Resample

Mit dem Werkzeug Resample können Sie die Größe des Bildes auf Kosten einer Interpolation aus der folgenden Liste ändern:

- Nächster Nachbar
- Bilinear
- Bikubisch
- Pixel zu Fläche Verhältnis
- Lanczos-4 (Standard)

Lanczos-4 ist das Verfahren, das die besten Ergebnisse liefert. Wenn Sie jedoch Artefakte sehen, insbesondere Sterne, die von schwarzen Pixeln umgeben sind, sollten Sie andere Interpolationsverfahren ausprobieren. Die Schaltfläche *Interpolation* wendet jedoch einen Begrenzungsfaktor auf die bikubische und Lanczos-4-Interpolation an, um Ringing-Artefakte zu vermeiden.

Wenn Sie das Bildverhältnis ändern möchten, sollten Sie die Schaltfläche *Seitenverhältnis beibehalten* deaktivieren.

Siril Kommandozeile

```
resample { factor | -width= | -height= | -maxdim= } [-interp=] [-noclamp]
```

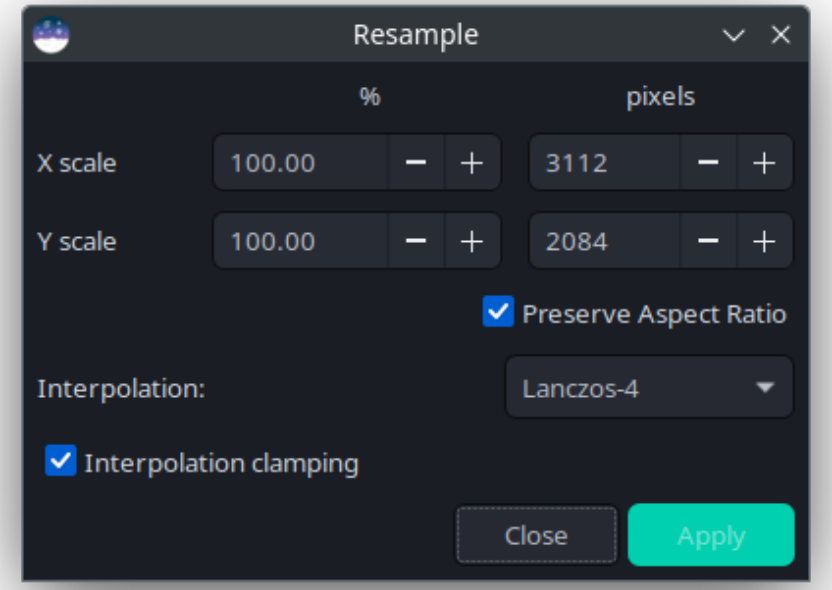


Abb. 75: Dialogbox Resample

Führt eine Neuskalierung des Bildes durch, entweder mit einem Faktor **factor** oder für die Zielbreite oder -höhe, die durch **-width=**, **-height=** oder **-maxdim=** vorgegeben wird. Dies wird im Allgemeinen verwendet, um die Größe von Bildern zu ändern: Ein Faktor von 0,5 teilt die Größe durch 2. Das Argument **-maxdim** kann verwendet werden, um die längste Dimension des Bildes auf eine festgelegte Größe zu ändern, was nützlich sein kann, um Bilder für bestimmte Websites zu optimieren, z. B. Social-Media-Websites.

In der grafischen Benutzeroberfläche können wir sehen, dass mehrere Interpolationsalgorithmen vorgeschlagen werden.

Die Pixelinterpolationsmethode kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **ne**[arest] (nächster), **cu**[bic] (kubisch), **la**[nczos4], **li**[near], **ar**[ea] (Bereich)}.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden

```
seqresample sequencename { -scale= | -width= | -height= } [-interp=] [-prefix=]
```

Skaliert die im Argument **sequencename** angegebene Sequenz. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Skalierungsfaktor wird entweder durch das Argument **-scale=** oder durch Festlegen der Ausgabebreite, -höhe oder -maximaldimension mit den Optionen **-width=**, **-height=** oder **-maxdim=** angegeben.

Eine Interpolationsmethode kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **ne**[arest] (nächster), **cu**[bic] (kubisch), **la**[nczos4], **li**[near], **ar**[ea] (Bereich)}. Bei der kubischen und der Lanczos-Interpolation wird die Klammerung angewendet.

Der Name der Ausgabesequenz beginnt mit dem Präfix "scaled_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

10.7 Hintergrund-Extraktion

Der Himmelshintergrund hat oft einen unerwünschten Farbverlauf bzw. Gradienten, der durch Lichtverschmutzung, den Mond oder einfach durch die Ausrichtung der Kamera relativ zum Boden verursacht wird. Diese Funktion tastet den Hintergrund an vielen Stellen des Bildes ab und sucht nach einem Trend in den Helligkeitsschwankungen und entfernt ihn anhand einer geglätteten Funktion, um zu vermeiden, dass Nebel mit entfernt werden.

Proben können automatisch platziert werden, indem Sie eine Dichte angeben (*Proben pro Zeile*) und auf *Generieren* klicken. Wenn Bereiche des Bildes um einen Faktor *Rastertoleranz* mal Sigma heller als der Median sind, wird dort keine Probe platziert.

Tipp

Wenn Sie sehr starke Gradienten haben, zum Beispiel bei Aufnahmen unter einem Stadthimmel mit hohem Bortle-Wert, reicht möglicherweise selbst die maximale Gittertoleranz nicht aus. In diesem Fall können Sie das Kontrollkästchen „Alle Proben behalten“ aktivieren und das gesamte Probenraster wird ausgefüllt. Anschließend müssen Sie manuell Proben von tatsächlichen astronomischen Objekten entfernen.

Nach der Generierung können Proben auch manuell hinzugefügt (Linksklick) oder manuell entfernt (Rechtsklick) werden.

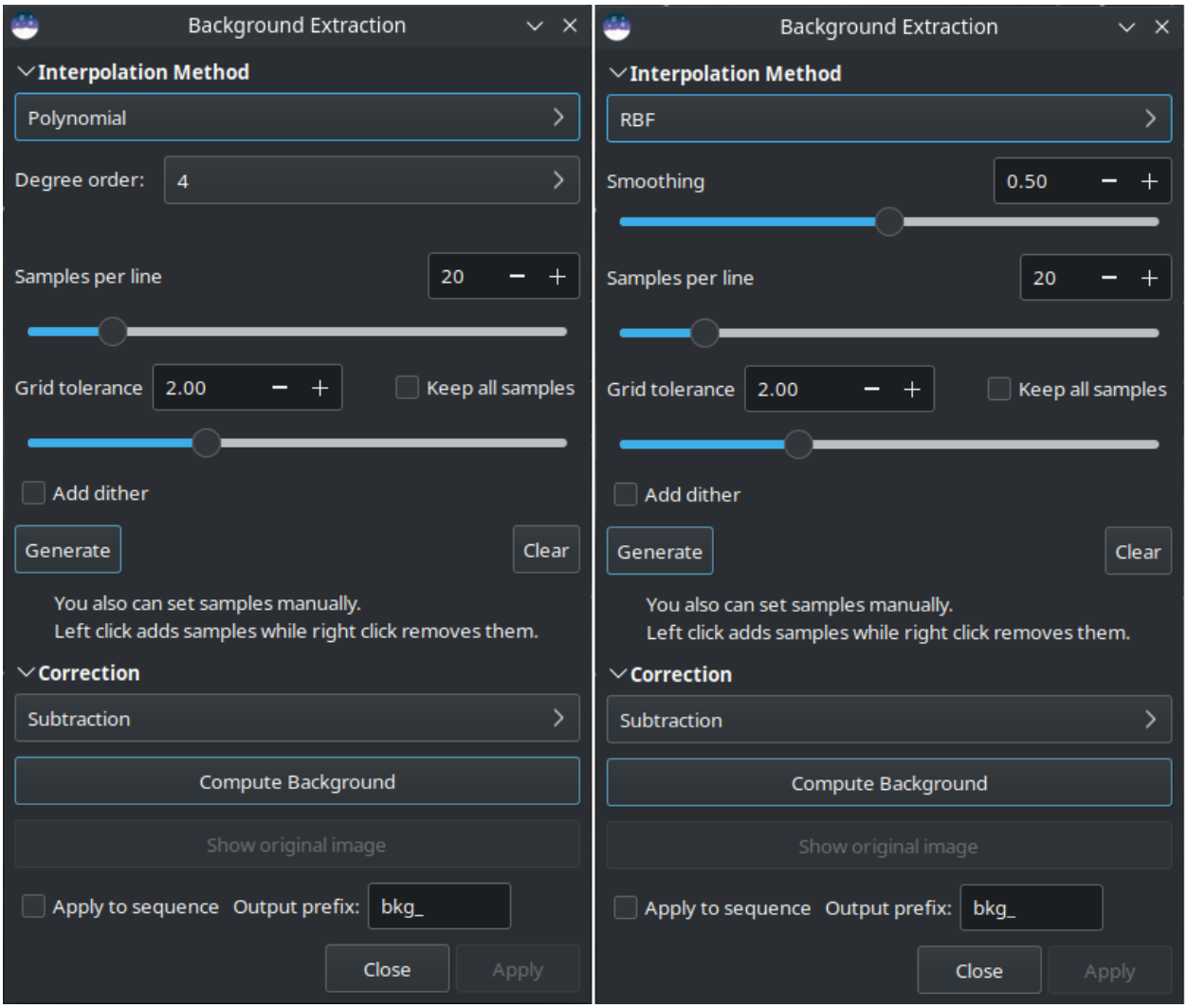


Abb. 76: Dialogbox Hintergrund-Extraktion. Auf der linken Seite ist die Polynom-Version zu sehen, rechts RBF.

Es gibt zwei Algorithmen zur Entfernung des Gradienten:

10.7.1 RBF

Dies ist die modernste Methode. Es verwendet die *Radiale Basisfunktion <https://de.wikipedia.org/wiki/Radiale_Basisfunktion>*, um einen Himmelshintergrund zu synthetisieren und den Farbverlauf mit großer Flexibilität zu entfernen. Es erfordert einen einzelnen Parameter, der in Form eines Schiebereglers vorliegt: *Glätten*. Mit diesem Wert können Sie bestimmen, wie weich oder hart der Übergang zwischen den Abtastpunkten berechnet wird. Bei großen und gleichmäßigen Verläufen ist ein hoher Glättungsfaktor sinnvoll, bei kleinen, lokalen Verläufen ein entsprechend niedrigerer Wert.

Tipp

Beginnen Sie mit der Grundeinstellung (50 %) und optimieren Sie diese schrittweise, um optimale Ergebnisse zu erzielen.

Theorie

Radiale Basisfunktionen sind Funktionen der Form $\phi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$, wobei wir in unserem Fall die euklidische Norm $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2}$ verwenden. Die Funktion f, die das Hintergrundmodell beschreibt, kann nun als lineare Kombination ausgedrückt werden

$$f(\mathbf{x}) = \sum_i w_i \, \phi(\|\mathbf{x} - \mathbf{x_i}\|) + o$$

wobei w_i den Gewichtungen für die verschiedenen Stichprobenpunkte entspricht und o einem konstanten Offset entspricht.

Die Anforderung, dass die Funktion f die Beispielpunkte durchlaufen soll, führt zu der Bedingung

$$\begin{pmatrix} \phi(\mathbf{x}_1 - \mathbf{x}_1) & \phi(\mathbf{x}_1 - \mathbf{x}_2) & \dots & \phi(\mathbf{x}_1 - \mathbf{x}_N) & 1 \\ \phi(\mathbf{x}_2 - \mathbf{x}_1) & \phi(\mathbf{x}_2 - \mathbf{x}_2) & \dots & \phi(\mathbf{x}_2 - \mathbf{x}_N) & 1 \\ \vdots & \vdots & \ddots & \vdots & \\ \phi(\mathbf{x}_N - \mathbf{x}_1) & \phi(\mathbf{x}_N - \mathbf{x}_2) & \dots & \phi(\mathbf{x}_N - \mathbf{x}_N) & 1 \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \\ o \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ 0 \end{pmatrix},$$

was nur erfüllt werden kann, wenn die Matrix auf der linken Seite invertierbar ist. Mit der richtigen Wahl der Funktion ϕ kann dies immer gewährleistet werden [Wright2003].

Zusätzlich wird der Summand $s\,I$ zur Matrix auf der linken Seite hinzugefügt, wobei s ein Glättungsparameter und I die Einheitsmatrix ist. Der Summand bewirkt eine Regularisierung, die zu einem glatteren Ergebnis führt, je größer der Parameter s ist. Dieser Parameter kann mit dem Parameter *Glätten* des Dialogfelds geändert werden.

Für die radiale Basisfunktion verwenden wir den Dünnplatten-Spline $\phi(|\mathbf{x}|) = |\mathbf{x}|^2 \log(|\mathbf{x}|)$.

10.7.2 Polynomisch

Dies ist der ursprüngliche und einfachste Algorithmus, der in Siril entwickelt wurde. Bei der Polynomberechnung wird nur ein Parameter verwendet: die *Ordnung*. Je höher der Ordnungsgrad, desto flexibler ist die Korrektur, aber ein zu hoher Grad kann zu seltsamen Ergebnissen wie einer Überkorrektur führen.

Tipp

Eine Korrektur der Stufe 1 kann sehr nützlich sein, wenn Sie den Farbverlauf auf den Subs/Lights entfernen möchten.

Warnung

Hintergrundentfernen kann auch an CFA-Bildern durchgeführt werden, aber nur an solchen, die ein Bayer-Muster haben. (Es ist nicht für X-TRANS Muster unterstützt) Bilder mit Bayer-Muster, werden als vier getrennte Bilder behandelt, jedes korrespondiert mit einem CFA-Unterkanal. Jeder Unterkanal wird unabhängig voneinander verarbeitet. Dann werden sie wieder in das originale Muster zusammengeführt.

Die vorgesehene Nutzung dafür ist das Entfernen von linearen Gradienten von Sequenz-Frames vor dem Anwenden von Drizzle auf eine Sequenz mit Bayer-Muster. In diesem Fall wird empfohlen, eine lineare (ploynomialer Grad 1) Gradient-Entfernung wie mit anderer Gradient-Entfernung, die vor dem Stacken angewendet wird.

10.7.3 Allgemeine Einstellungen

- **Dithering hinzufügen**: Aktivieren Sie diese Option, wenn nach der Hintergrundextraktion Farbstreifen entstehen. Dither ist eine bewusst angewandte Form des Rauschens, die verwendet wird, um Quantisierungsfehler zu randomisieren und so großflächige Muster wie Farbstreifen in Bildern zu verhindern.
- **Korrektur**:
  - **Subtraktion**: Sie wird hauptsächlich zur Korrektur von additiven Effekten verwendet, z. B. von Gradienten, die durch Lichtverschmutzung oder den Mond verursacht werden.
  - **Division**: Sie wird hauptsächlich zur Korrektur multiplikativer Phänomene verwendet, wie z. B. Vignettierung oder differentielle atmosphärische Absorption. Diese Art von Operation sollte jedoch besser mit einer Master-Flat-Korrektur durchgeführt werden.
- **Berechne den Hintergrund-Gradienten**: Dies berechnet den synthetischen Hintergrund und wendet die ausgewählte Korrektur an. Das Modell wird immer aus dem im Speicher geladenen Originalbild berechnet, so dass der Benutzer iterativ arbeiten kann.
- **Zeige Originalbild**: Halten Sie diese Schaltfläche gedrückt, um das Originalbild zu sehen.

Der Hintergrundgradient eines vorverarbeiteten Bildes kann komplex sein, da sich der Gradient mit der Aufnahmesitzung gedreht haben kann. Es kann schwierig sein, ihn vollständig zu entfernen, da es schwierig ist, ihn mit einer Polynomfunktion darzustellen. Wenn dies der Fall ist, können Sie in Erwägung ziehen, den Gradienten in den Einzelframes zu entfernen: In einem Einzelbild ist der Hintergrundgradient viel einfacher und folgt im Allgemeinen einer einfachen linearen Funktion (Grad 1).

Tipp

Manchmal treten nach der Hintergrundextraktion unschöne Farbstreifen auf. In diesem Fall müssen zwei Dinge überprüft werden. Erstens: Wenn es sich um ein 16-Bit-Bild handelt, empfehlen wir Ihnen dringend, immer das 32-Bit-Format zu verwenden. Wenn Sie trotz allem immer noch solche Artefakte beobachten, ist die oben erläuterte Option **Dithering hinzufügen** die Lösung für Ihr Problem.

Abb. 77: Wenn eine solche Streifenbildung nach der Gradientenextraktion auftritt, kann dies mit der Option **Dithering hinzufügen** behoben werden (mit freundlicher Genehmigung von Nathan B.).

Tipp

Gute Ergebnisse mit dem RBF-Algorithmus erfordern im Allgemeinen weniger Stichproben als mit dem Polynomalgorithmus.

Siehe auch

Weitere Erklärungen finden Sie in der entsprechenden Anleitung hier (englisch).

Siril Kommandozeile

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-
↪existing]
```

Berechnet einen synthetischen Hintergrundgradienten unter Verwendung des Polynomfunktionsmodells des Grades **degree** oder des RBF-Modells (wenn stattdessen **-rbf** angegeben wird) und subtrahiert ihn vom Bild.

Die Anzahl der Stichproben pro horizontaler Linie und die Toleranz zum Ausschluss hellerer Bereiche können mit

den optionalen Argumenten angepasst werden. Die Toleranz wird in MAD-Einheiten angegeben: Median + Toleranz * mad.

Dithering, das für geringe dynamische Gradienten erforderlich ist, kann mit **-dither** aktiviert werden.

For RBF, the additional smoothing parameter is also available. To use pre-existing background samples (e.g. if you have set background samples using a Python script) the **-existing** argument must be used

Tipp

The -existing command argument forces use of existing background samples. This option is primarily for use in conjunction with the Python module where `SirilInterface.set_bgsamples()` can be used to set custom background samples based on user-defined algorithms. If it is not provided, *subsky* will automatically regenerate background samples. Note that the -existing option is not available with the seqsubsky command, because sequence frames are not necessarily registered at the time background subtraction is carried out, so the samples for one frame do not necessarily apply to another.

Siril Kommandozeile

```
seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-
↪smooth=0.5] [-prefix=]
```

Gleicher Befehl wie SUBSKY, aber für die Sequenz **sequencename**.

Dithering, required for low dynamic gradients, can be disabled with **-nodither**. Note that the **-existing** option is not available for sequence background removal, as the frames of a sequence are not necessarily always aligned.

Der Name der Ausgabesequenz beginnt mit dem Präfix "bkg_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde. Nur ausgewählte Bilder der Sequenz werden verarbeitet

Verweis: *subsky*

10.8 GraXpert Schnittstelle

Tipp

The first GraXpert interface was written in C and simply called the GraXpert executable. However for various reasons this approach proved unreliable for a significant minority of users and with Siril 1.4.0-beta2 an updated interface has been released as a Python script. This directly interfaces with the GraXpert model files and provides much better performance and integration, and since it builds on the way the Python interface uses the native PyPI modules library to install the correct versions of dependecies for each user's machine it is expected to be more reliable and better able to make use of different types of GPU.

Owing to the way GraXpert distributes its AI models from an Amazon S3 instance, a copy of the GraXpert program needs to be available in order to download model files. The model files you can download are dependent on the version of GraXpert installed therefore it is recommended to install 3.1.0-RC2 which provides access to the stellar and background object deconvolution models as well as the denoising and background extraction models that have been available since version 3.0.0.

The old GraXpert interface was marked as deprecated in 1.4.0-beta2 and has since been removed from the code base. This documentation covers the new Python-based interface.

Siril provides an interface to the AI models provided by the third party GraXpert tool. This is a third party piece of Free and Open Source software that provides sophisticated background removal and denoising routines. There is a lengthy history of collaboration between Siril and GraXpert, and the RBF algorithm used in Siril's core background removal function was contributed by the GraXpert developers several years ago.

In order to be used, the path to the working GraXpert installation must be set in *Preferences->Miscellaneous*.

10.8.1 Installation und Konfiguration

The GraXpert-AI.py script can be installed in the usual way, by going to the *Scripts* menu -> *Get Scripts* and selecting the check box next to the script in the list.

To configure the script you need to tell Siril where your GraXpert installation is. To do this, go to *Preferences* -> *Miscellaneous* and select the GraXpert executable using the appropriate file chooser widget.

10.8.2 General Aspects of the GUI

Die GUI wird unten angezeigt:

The title bar changes to show the currently selected operation, and at the top of the window are the script title and version number. A dropdown allows you to select the operation you wish to perform. How this dropdown is populated depends on the version of GraXpert you have available. It will always feature Denoising and Background Extraction, even if no locla models are available (see the section on Model Manager for information on how to download models). If you have locally available deconvolution models, or if you have set the location of a GraXpert version that supports deconvolution, you will also see Deconvolution (Stars) and Deconvolution (Objects), however if these models are not available locally and you don't have a version of GraXpert that is capable of downloading them they will not be shown.

Next follows an area for operation-specific parameters. If you select Denoising you will find a widget to control strength, if you select Background Extraction you will find widgets for smoothing and the type of background extraction to perform, and so on. These are described in more detail below.

The Advanced setting provides a means of controlling the inferencing batch size and whether or not to use GPU acceleration. It is generally a good idea to leave the batch size at the default: it sets how many chunks of the image can be processed in parallel, and increasing it may result in out-of-memory errors.

Tipp

The GraXpert models use ONNX to perform inferencing. This is a hardware- agnostic Machine Learning framework that provides various runtimes targeting different types of acceleration hardware. There is a specific NVidia runtime, a DirectML runtime that provides generic Windows GPU acceleration, a MacOS runtime and Linux runtimes for NVidia, AMD and Intel GPUs. The script will automatically try to install the most suitable runtime for your hardware, however in the event that it does not work there is always a CPU runtime to revert to - simply disable the GPU acceleration checkbox and please submit a bug report as this is new functionality and we are especially keen to hear of any issues relating to the ONNX runtime selection.

Finally, the UI provides *Apply* and *Close* buttons that perform the expected functions and a *Model Manager* button that opens a tool to discover and download versions of the AI models.

When opening the GUI, it will detect whether a single image or a sequence is loaded. If a single image is loaded it will process it and update the result in Siril, whereas if a sequence is loaded it will process all the selected frames of the sequence and create a new sequence with a prefix dependent on the operation being performed.

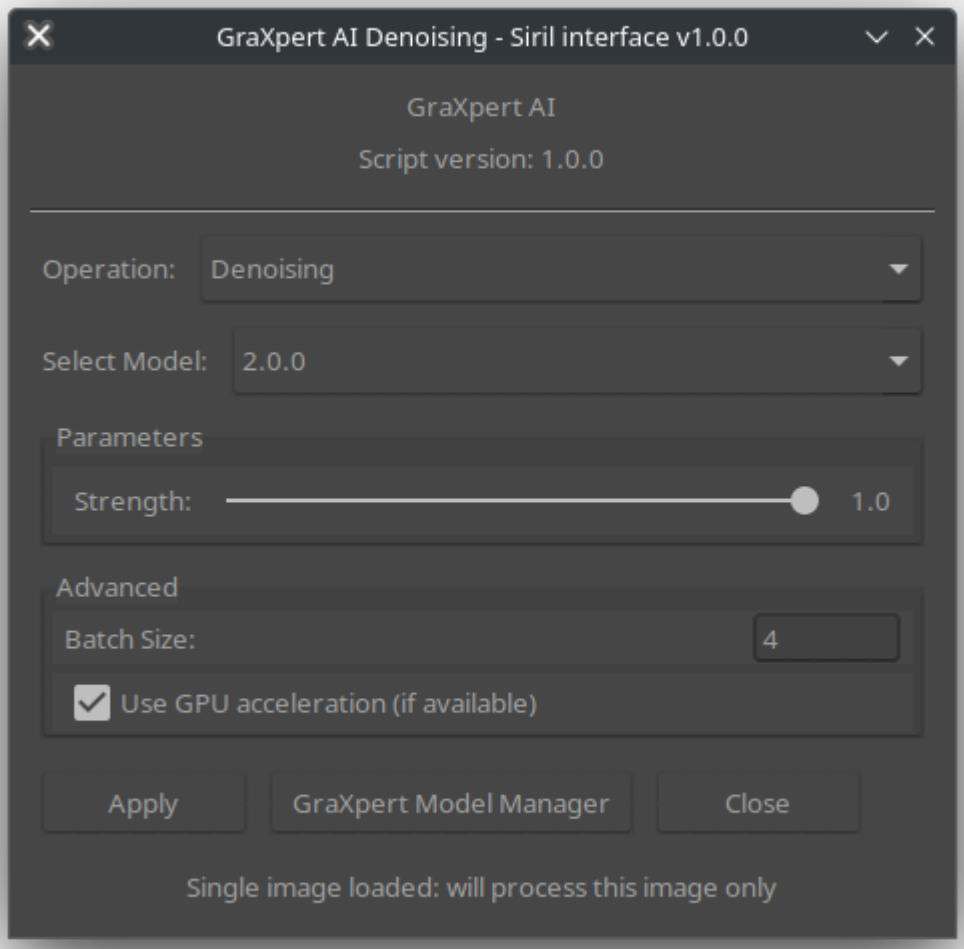


Abb. 78: GraXpert script interface, showing denoising options.

Warnung

In Siril 1.4, Python scripting cannot subscribe to Siril event notifications so it is not possible for the script to detect if the image has been closed or if a change between a single image and a sequence occurs after opening the tool. Therefore if you do this, you should close and reopen the script GUI.

10.8.3 Hintergrund-Extraktion

The GraXpert background extraction AI model offers two parameters:

- Smoothing - this determines the smoothness of the background model that is generated.
- Correction type - this determines the type of correction that is applied: you can choose from subtraction or division.

Note that the GraXpert script does not provide an interface to the classical background removal algorithms provided by GraXpert. RBF is already natively available in Siril, and Spline and Kriging do not generally provide better performance than RBF.

10.8.4 Dekonvolution

From version 3.1 onwards, GraXpert supports deconvolution. With the models downloaded, Siril supports both modes provided by GraXpert:

- Objekt Dekonvolution
- Sternentfaltung (Dekonvolution)

Diese Modi können über das Dropdown-Menü *Operation* ausgewählt werden.

Both deconvolution modes have two parameters, *Strength*, which takes values between 0.0 (no deconvolution) and 1.0 (maximum deconvolution) and *Image FWHM (in pixels)*, which also takes values between 0.0 (prioritises fine structure retention) and 10.0 (prioritise deconvolution ability). If you set it to zero you can see a very (too) small effect on processed image, if so set it higher.

10.8.5 Entrauschung

To use the GraXpert AI denoising model, select *Denoising* in the Operations dropdown. This operation has only one parameter, *Strength*, which takes values between 0.0 (no denoising) and 1.0 (maximum denoising).

Denoising can take a little time, especially for large images and / or if using a CPU for inferencing. However once the denoising is done the denoised image is cached, so experimenting with different strengths is extremly quick.

10.8.6 Modell Manager

The script provides a model manager to discover and download model versions. Whereas inferencing is done directly in the script, model download requires use of the actual GraXpert program because the models are stored in an Amazon S3 instance and GraXpert stores its credentials internally. The model manager will show radio buttons for each model that the installed version of GraXpert is capable of downloading. For v3.0.x this means denoising and background extraction; for v3.1.0-rcx this also adds stellar and object deconvolution.

The list of remotely available models is shown. If you want to discover models for a different AI operation, change the radio button to the operation you want a model for and click *Refresh*. A model can be downloaded by selecting it in the list and clicking *Download selected model*. Note that the download may take some time in the case of large models or slow network connections and GraXpert provides little feedback, so be patient. The models are stored in the GraXpert user data directory.

Warnung

Do not move the GraXpert models from the GraXpert data directory structure where they are saved by the model manager. This is where the script expects to find them and it will ignore models put in any other location.

Once a model is downloaded it will automatically be populated in the model version dropdown in the main UI when the relevant AI operation is selected.

10.8.7 Skripting

The GraXpert-AI.py script provides a command-line interface for use in scripting. This uses the *pyscript* command as shown below.

Siril Kommandozeile

```
pyscript [-async] scriptname.py [script_argv]
```

Executes a Siril python script

An optional argument **-async** may be provided. If it is provided it **must** be the first argument. This allows the script launched by pyscript to run in the background without blocking a script that calls it. This is expected to be useful for launcher-type scripts. The default is for the script to block until completion if pyscript is run as part of a script, but not to block if run from the GUI command entry (so the -async argument makes no difference when running like that).

The script name must be provided next (or first, if **-async** is not provided). If it is not found in the current working directory, the user-defined script paths specified in Preferences and the local siril-scripts repository will be searched. All subsequent arguments will be treated as script arguments and passed to the script as its argument vector. Note that the specific script must incorporate support for reading input from the argument vector

Betriebsparameter

Die folgenden Parameter werden verwendet, um die auszuführende Operation anzugeben:

- **-denoise** specifies denoising
- **-deconv_obj** specifies object deconvolution
- **-deconv_stellar** specifies stellar deconvolution
- **-bge** spezifiziert die Hintergrund-Extraktion

allgemeine Parameter

Die folgenden Parameter werden in allen Modi erkannt:

- **-h / --help** prints help to the Siril console and exits
- **-model** specifies the model version to use (e.g. -model="3.0.2") If this argument is omitted it defaults to the latest available version.
- **-batch** specifies the AI batch size
- **-gpu** specifies to use the GPU (this is the default)

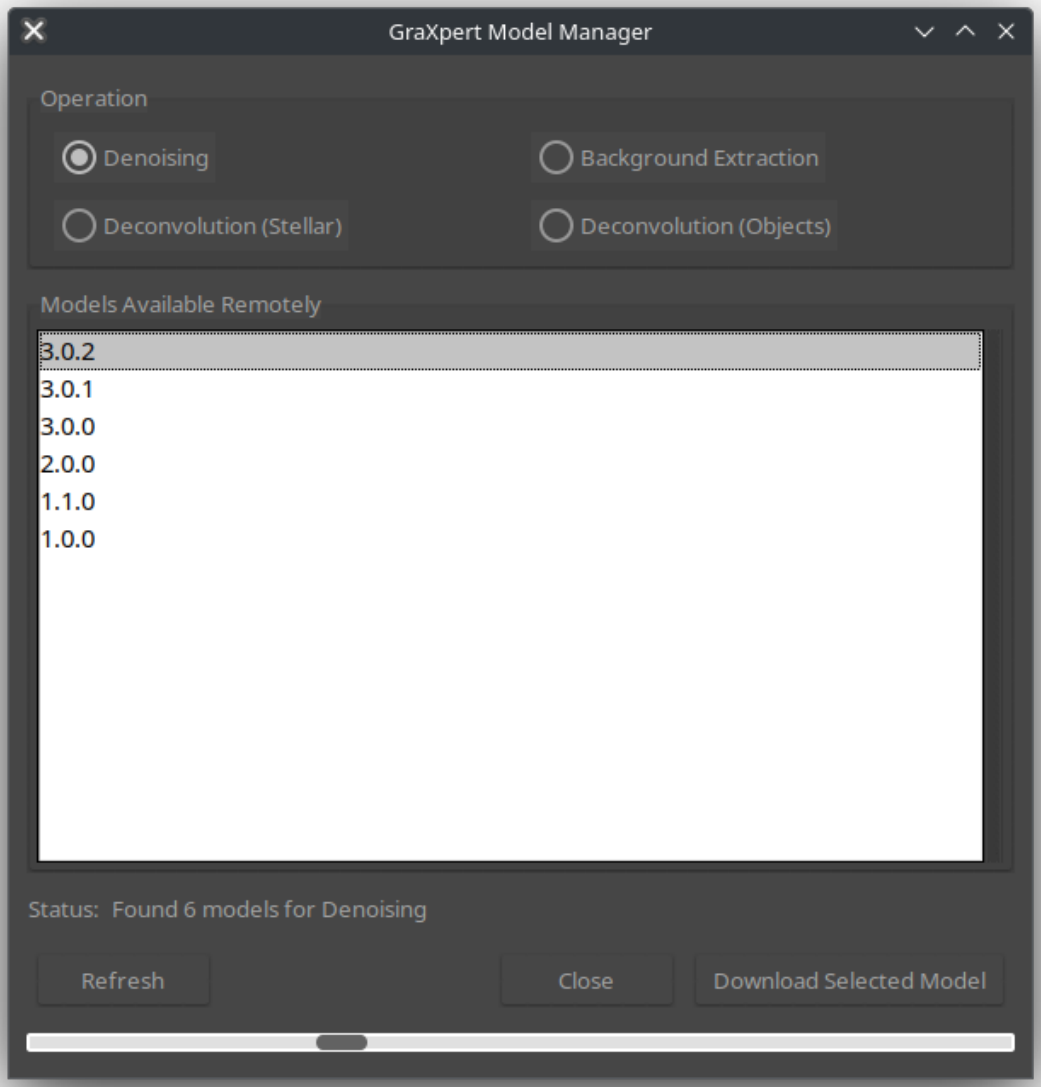


Abb. 79: GraXpert script interface, showing the model manager.

- **-nogpu** spezifiziert Arten, die nur CPU-Inferenz verwenden
- **-listmodels** causes the script to list the available model versions for the specified operation and exit

> **Tipp**
>
> Note that this script does not provide a -seq parameter as it expects to work on a sequence already loaded in Siril. When scripting, this can be achieved using the *load_seq* command first to load the desired sequence.

Parameter für die Hintergrundextraktion

- **-smoothing=** This option sets the smoothing parameter, between 0.0-1.0
- **-correction=** This option sets the correction type and may be either "subtraction" or "division"
- **-keep-bg** This option specifies that the modelled background should be saved as well. For a filename such as myfile.fit, the background will be saved as myfile_bg.fit

Entrauschungsparameter

- **-strength=** legt die Stärke zwischen 0,0 und 1,0 fest

Dekonvolutionparameter

Die gleichen Parameter stehen sowohl für die Stern- als auch für die Objekt-Dekonvolution zur Verfügung:

- **-strength=** legt die Stärke der Deconvolution zwischen 0,0 und 1,0 fest
- **-psfsize** This option sets the PSF size, as described above

Beispiele

pyscript GraXpert_AI.py -denoise -model='3.0.1'

Or, if calling it from another Python script:

siril.cmd("pyscript", "GraXpert-AI.py", "-denoise", "-model='3.0.1'")

10.9 Kanalextraktion

10.9.1 Farbkanäle aufteilen

This function creates three monochrome images from a 3-channel color image, depending on the configured color space. For RGB, it's simply splitting the file in three. For the others, it involves computation of the equivalent color space, either HSL (hue-saturation-lightness), HSV (hue-saturation-value) or CIELAB.

> **Tipp**
>
> Wird kein Name für einen Kanal angegeben, wird der Kanal nicht extrahiert.

> **Siril Kommandozeile**

```
split file1 file2 file3 [-hsl | -hsv | -lab]
```

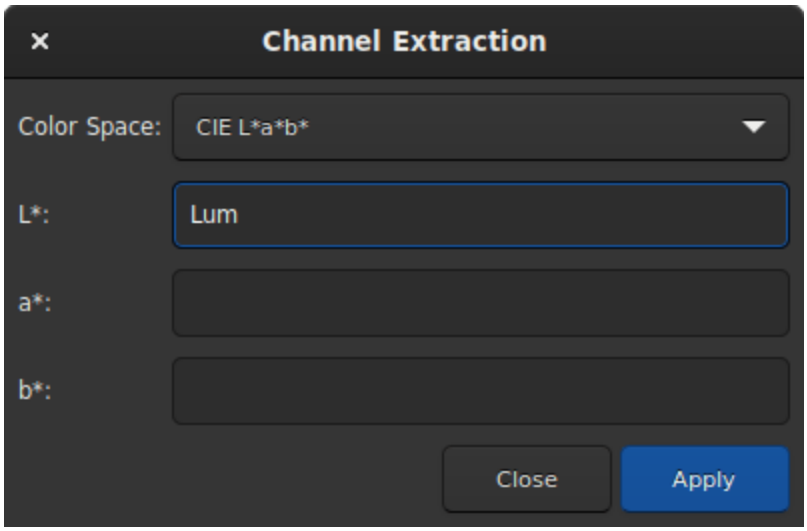


Abb. 80: Dialogbox Kanal Extraktion.

Teilt ein Farbbild in drei verschiedene Dateien auf (eine für jede Farbe) und speichert sie in den Dateien **file1**.fit, **file2**.fit und **file3**.fit. Als letztes Argument kann optional angegeben werden, **-hsl**, **-hsv** oder **lab**, um eine HSL-, HSV- oder CieLAB-Extraktion durchzuführen. Wenn keine Option angegeben wird, erfolgt die Extraktion im RGB-Format, d.h. es wird keine Konvertierung durchgeführt

10.9.2 CFA aufteilen

CFA bedeutet Farbfilteranordnung. Dieser Begriff wird häufig verwendet, um den einkanaligen Bildinhalt eines Farbbildes zu beschreiben, wobei jedes Pixel den Werten entspricht, die hinter einem sensoreigenen Farbfilter erfasst wurden. Dies steht im Gegensatz zu debayerten Bildern (oder Debayered oder Demosaic).

Das Öffnen eines CFA-Bildes in Siril ist für die Vorverarbeitung/Kalibrierung erforderlich, z. B. das Entfernen des Dunkelstroms vor der Interpolation des Bildes in ein 3-Kanal-Farbbild. Wir können die Funktion "CFA aufteilen" verwenden, um diese Daten zu extrahieren:

- **CFA (Color Filter Array) Kanäle aufteilen**: Aus dem CFA-Bild werden vier Bilder erstellt, die jeweils einen Filter der Bayer-Matrix darstellen, also im Allgemeinen R.fit, G1.fit, G2.fit und B.fit. Dies ist nützlich, wenn das Ziel darin besteht, die verschiedenen Farben des Bildes getrennt zu verarbeiten.
- **Ha Extrahieren**: Die Verwendung eines H-Alpha-Filters mit einem Farbkamerabild (OSC: on-sensor color, oder one-shot color camera) bedeutet, dass nur die Pixel mit Rotfilter nützlich sind, also im Allgemeinen nur ein Viertel davon. Diese Funktion erzeugt ein neues Bild, das nur die Pixel enthält, die mit dem Rotfilter verbunden sind, der in der Bayer-Matrix des Bildes dokumentiert ist.
- **Ha/OIII extrahieren**: Für OSC-Kameras gibt es Filter, die (fast) nur Photonen der Wellenlängen H-alpha und O-III durchlassen. Bei dieser Extraktion werden zwei Bilder erzeugt: ein Bild aus den roten Pixeln wie bei Extract Ha und ein Bild, das die grünen und blauen Pixel zu einem Bild für O-III kombiniert. Beide Bilder entsprechen der halben Auflösung des Eingabebildes.

Bemerkung

Es gibt eine häufig gestellte Frage dazu, warum Ha- und OIII-Bilder unterschiedliche Größen haben und wie sie aufgeteilt werden. Dieser Hinweis versucht, eine Antwort auf diese Frage zu geben.

In einem Farbbildsensor sind die Pixel mit einer sehr feinen Filtermatrix bedeckt, die als Farbfilterarray (CFA) oder Bayer-Matrix bezeichnet wird. Die Anordnung der gefilterten Pixel ist eines von mehreren Mustern: RGGB, GBGR usw.

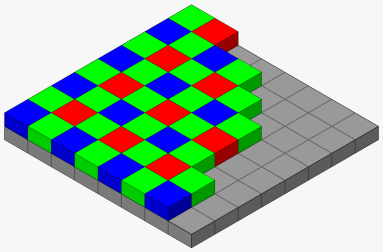


Abb. 81: Originalbild by Cburnett, licensed as CC BY-SA 3.0.

Von diesen Pixeln reagieren nur die R-Pixel auf Ha. Daher teilen wir zunächst alle roten Pixel in ein Ha-Bild auf. Da nur 1 von 4 CFA-Elementen rot ist, sind die Bildabmessungen des Ha-Bilds halb so groß wie die des ursprünglichen Einzelbildes.

Die übrigen Pixel G und B sind alle empfindlich gegenüber OIII. Die Empfindlichkeit der grünen Pixel gegenüber OIII unterscheidet sich von der Empfindlichkeit der blauen Pixel gegenüber OIII, sie bilden jedoch dieselbe Szene ab und sind gleichmäßig verteilt, sodass die durchschnittliche Intensität dieselbe sein muss.

$$\mathrm{G_i} = \mathrm{G_{io}} \times \frac{3 \times \overline{\mathrm{G_o}}}{2 \times \overline{\mathrm{G_o}} + \overline{\mathrm{B_o}}}$$
$$\mathrm{B_i} = \mathrm{B_{io}} \times \frac{3 \times \overline{\mathrm{B_o}}}{2 \times \overline{\mathrm{G_o}} + \overline{\mathrm{B_o}}}$$

Wobei $\mathrm{B_i}$ ist: i^{th} blue pixel, $\mathrm{B_{io}}$ ist: i^{th} Original-Blau und $\overline{\mathrm{B_o}}$ ist der Durchschnitt aller blauen Pixel (und analog für grüne Pixel).

Bisher haben wir einen ausgeglichenen Satz von G- und B-Pixeln mit Lücken, wo die R-Pixel entfernt wurden. Abschließend verwenden wir eine bilineare Interpolation, um die R-Pixelwerte abzuschätzen und erhalten ein OIII-Bild in voller Größe.

Bemerkung

Die Ha/OIII-Resampling-Option bestimmt, wie die Ausgabe von Extract Ha/OIII behandelt wird. Keine Neuabtastung erzeugt ein OIII-Bild mit voller Auflösung und ein Ha-Bild mit halber Auflösung; Upsample Ha vergrößert das Ha-Bild um den Faktor 2, um es dem OIII-Bild anzupassen; Downsample OIII verkleinert das OIII-Bild um den Faktor 2, um es dem Ha-Bild anzupassen.

Möglicherweise möchten Sie Drizzling verwenden, um die Ha-Daten statt des Hochskalierens hochzurechnen. Da Drizzling eine Stackingmethode ist, müssen Sie in diesem Fall *seqextract_HaOiii* verwenden, um Ha und OIII aus jedem Frame der Sequenz zu extrahieren und dann die OIII-Bilder auf die übliche Weise und die Ha-Bilder mit einem 2x Drizzling stacken.

- **Grün extrahieren**: Für die Fotometrie ist es oft nützlich, nur den grünen Teil des CFA-Bildes zu verarbeiten, da dieser empfindlicher ist und zwei Pixel zum Mitteln hat, was das Rauschen noch weiter reduziert. Natürlich wird auch die Schärfe des erzeugten Bildes um die Hälfte reduziert.

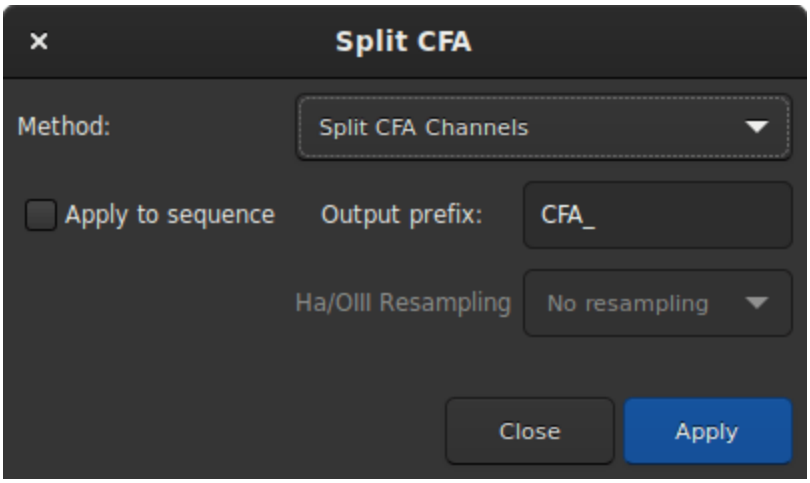


Abb. 82: Dialogbox CFA aufteilen.

Bemerkung

Diese Funktionen funktionieren nur, wenn die Bayer-Matrix von der Aufnahmesoftware ordnungsgemäß dokumentiert wurde und wenn das Bildformat sie unterstützt, also im Allgemeinen FITS oder SER.

Warnung

Dies funktioniert nicht mit anderen Filtermatrizen als den Bayer-Matrizen, wie dem Fujifilm X-TRANS.

10.9.3 Wavelet Layer

Dieses Tool extrahiert die verschiedenen Ebenen des Bildes durch Anwendung des Wavelet-Verfahrens. Jede Ebene wird in einem Bild gespeichert und der Satz von Bildern kann als Sequenz gelesen werden. Sie können bis zu 9 Ebenen für die Wavelet-Berechnung wählen, und der Algorithmus ist entweder linear oder BSpline. Letzteres ist in der Regel die bevorzugte Methode.

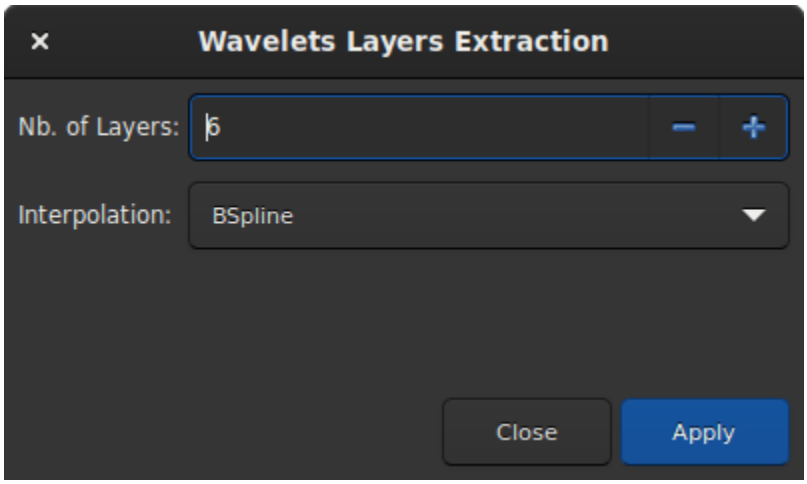


Abb. 83: Dialogfeld zur Extraktion von Wavelet-Ebenen.

Die Zerlegung erfolgt durch eine Reihe von Detailschichten, die mit zunehmenden charakteristischen Maßstäben definiert werden, und eine abschließende Restschicht, die die verbleibenden unaufgelösten Strukturen enthält.

10.10 Lineare Übereinstimmung

Beim linearen Abgleich geht es darum, eine lineare Funktion zu finden, die die Intensität der Pixel eines Bildes am besten (im Sinne der kleinsten Quadrate) mit denen eines Referenzbildes abgleicht. Dies ist eine schnelle und einfache Methode, um die Histogramme verschiedener Bilder einander anzugleichen.

Mit dem *Referenz-Bild* können Sie das Referenzbild auswählen.

Die Schieberegler *Ausschluss Low* und *Ausschluss High* ermöglichen es, Pixelwerte am linken und rechten Rand der Intensitätsverteilungen auszuschließen. Sie sind als Quantile definiert, die im Bereich [0, 1] liegen. Der Standardwert für hoch ist beispielsweise 0,92, was bedeutet, dass die 8% hellsten Pixel von der Anpassung ausgeschlossen werden, um die linearen Anpassungskoeffizienten zu finden.

Warnung

Das Bild und die Referenz müssen vor der Anwendung eines linearen Abgleichs registriert werden. Andernfalls gibt es keinen Grund für die Annahme, dass die Intensitäten der Pixel korreliert sind.

Siril Kommandozeile

```
linear_match reference low high
```

Berechnet eine lineare Funktion zwischen einem **reference**-Bild und dem geladenen Bild im Speicher und wendet sie an.

Der Algorithmus ignoriert alle Referenzpixel, deren Werte außerhalb des Bereichs [**low**, **high**] liegen

10.11 RGB-Zusammensetzung

Mit dem Werkzeug RGB-Zusammensetzung können Sie bis zu 8 monochrome Bilder zu einem einzigen Farbbild zusammensetzen. Das Werkzeug wurde aktualisiert und bietet nun Zugang zu den besten Registrierungsmethoden, die in Siril verfügbar sind, sowie zu einer manuellen Registrierungsmethode, die es nur in diesem Tool gibt.

Die Bedienung dieses Werkzeugs ist recht einfach: Laden Sie einfach die Bilder und weisen Sie ihnen eine Farbe zu. Das erste Feld, das optional ist, ist für die Luminanzebene reserviert. Sobald eine Luminanzebene geladen ist, können Sie sie dank der Schaltfläche *Luminanz verwenden* in die Komposition integrieren oder nicht. Jede Farbe kann angepasst werden, indem man auf sie klickt und eine neue Farbe auswählt. Wenn mehr als 3 Bilder (oder 4, wenn Luminanz vorhanden ist) geladen sind oder wenn die zugewiesenen Farben Farbmischungen und nicht reine R-, G- und B-Farben sind, kann es notwendig sein, die Helligkeit jedes Kanals anzupassen. Die Schaltfläche *Helligkeit der Ebenen anpassen* führt diesen Vorgang automatisch durch.

When compositing different channels, especially from narrowband filters, the histograms may be severely mismatched. A basic initial correction can be made by applying a linear match using the *Linear match* button. This works in the same way as the *Image Processing* menu entry, but is tailored to this particular tool. The top layer loaded (usually Luminance or Red) is taken as the reference channel, and all other channels are automatically linearly matched to it. This is particularly important if assigning non-pure colours to some layers, as each layer will potentially be blended

Abb. 84: Originalbild von M45 (courtesy of V. Cohas).

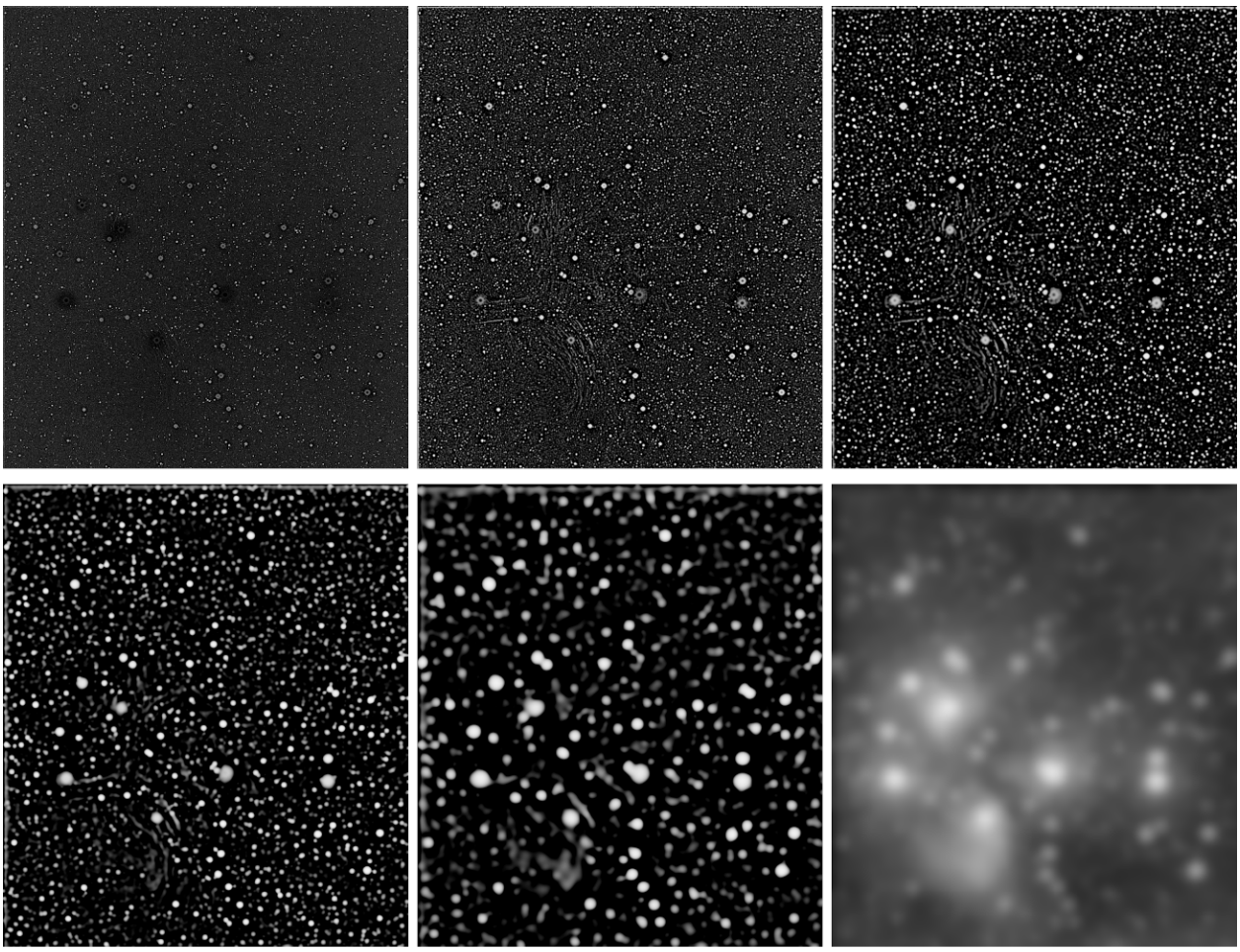


Abb. 85: 6 extrahierte Ebenen.

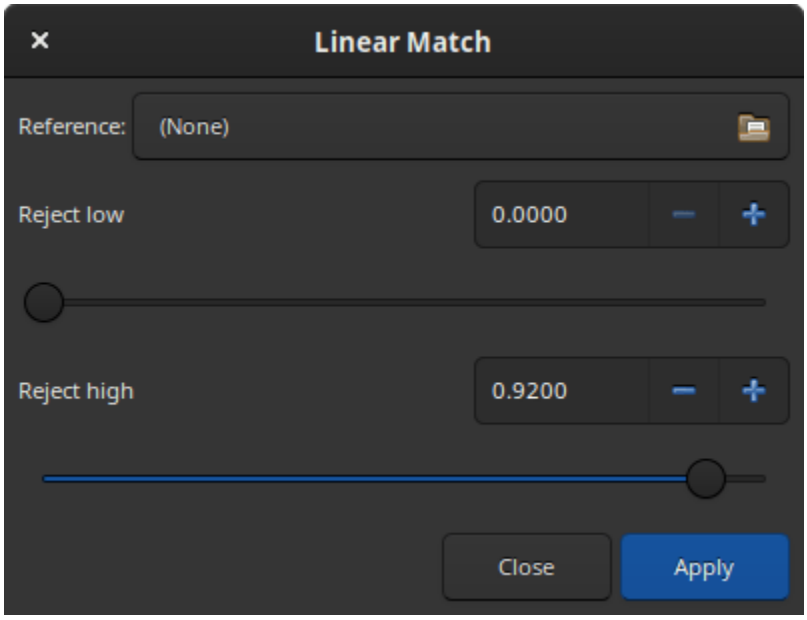


Abb. 86: Dialogfeld Lineare Übereinstimmung

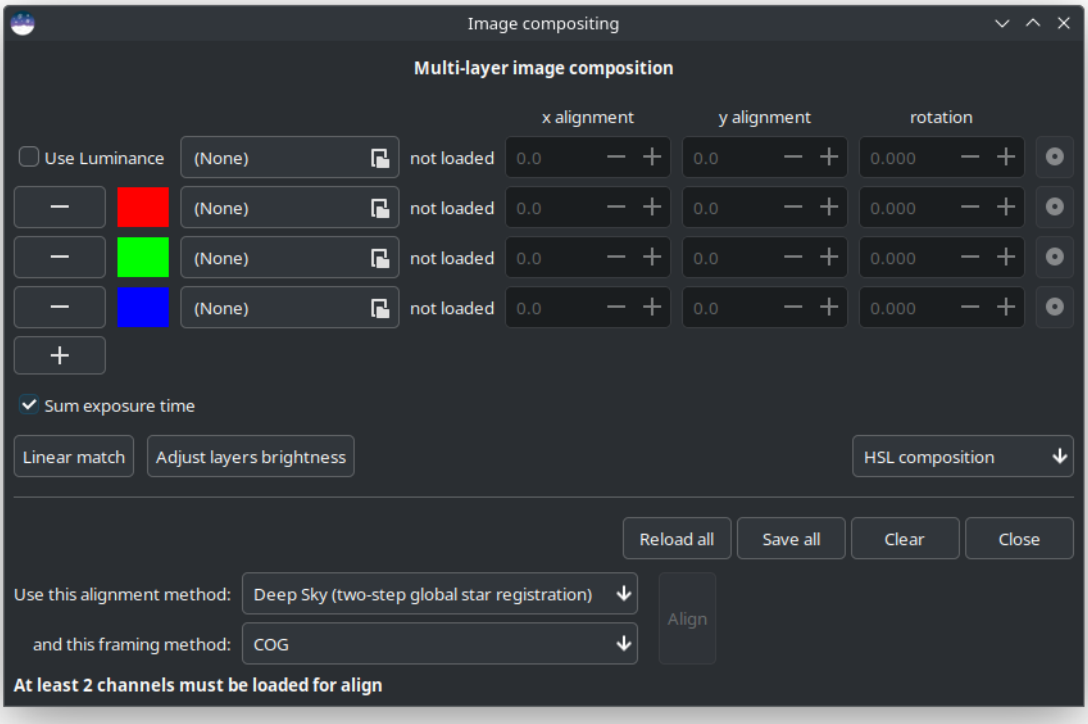


Image compositing
Multi-layer image composition
x alignment
y alignment
rotation
Use Luminance
(None)
not loaded
0.0
0.0
0.000
(None)
not loaded
0.0
0.0
0.000
(None)
not loaded
0.0
0.0
0.000
(None)
not loaded
0.0
0.0
0.000
Sum exposure time
Linear match
Adjust layers brightness
HSL composition
Reload all
Save all
Clear
Close
Use this alignment method:
Deep Sky (two-step global star registration)
and this framing method:
COG
Align
At least 2 channels must be loaded for align

into each RGB channel in the result, and it will no longer be possible to separately adjust the histograms of the input layers by adjusting the R, G and B histograms of the composition.

Es ist möglich, die Option *Belichtungszeit summieren* zu aktivieren oder nicht. Diese Option gibt dem Benutzer die Möglichkeit, die Belichtungszeiten der einzelnen Bilder zu addieren oder nicht und das Ergebnis im FITS-Header zu speichern.

Bemerkung

Für das Binning und die Bildgröße gilt, dass das zuerst geladene Bild die Größe des Ausgabebildes bestimmt. Wenn Sie Bilder unterschiedlicher Größe haben, sollten Sie immer das größte Bild zuerst laden. Wenn Ihre Bilder nur wegen des Binnings unterschiedlich groß sind, also dasselbe Sichtfeld haben, skaliert das Kompositionswerkzeug die kleineren Bilder hoch, wenn sie geladen werden, damit sie der Größe des ersten geladenen Bildes entsprechen. Das ist nützlich für die übliche L-RGB-Komposition, bei der mit den Farbfiltern in Bin 2 aufgenommen wurde. Das bedeutet auch, dass es unwahrscheinlich ist, dass zwei Bilder, die nicht mit demselben Sensor aufgenommen wurden, nach dem Resampling das gleiche Sichtfeld und die gleiche Pixelabtastung haben. Ob dieses Werkzeug solche Bilder abgleichen kann, hängt von der Skalenabweichung und der verwendeten Abgleichsmethode ab.

Für das Rendern der Komposition stehen drei Farbräume zur Verfügung:

- HSL (for Farbton (hue), Sättigung (saturation), Helligkeit (lightness))
- HSV (für Farbton/hue, Sättigung/saturation, Wert/value; auch bekannt als HSB, für Farbton/hue, Sättigung/saturation, Helligkeit/brightness)
- CIE L*a*b*

und sind der Wahl des Benutzers überlassen.

Once the composition is finished, it is possible to do a linear match of the channels using the *Linear match* button.

Wenn die Bilder nicht aneinander ausgerichtet sind, ist es möglich, sie auszurichten. Die folgenden Algorithmen sind möglich:

- **Deep Sky (two step global star registration)**: no selection is required for this method. This generates a very accurate 8 degrees-of-freedom homography mapping between images.
- **Planetary (DFT pattern alignment)**: you have to draw a selection around the object you want to align. A high enough contrast is required for the algorithm to work properly. This method only registers image shifts, it cannot compensate for rotation. This is fine for planetary images but for deep sky images it is recommended to use the global star registration method.
- **KOMBAT**: as with the DFT pattern matching, you have to draw a selection around the object you want to align. This method also only registers image shifts. It is much faster than the DFT method, but can be less successful when the image patterns in the different channels are significantly different (for example, two planetary channels filtered with IR and UV filters may not share enough pattern similarity for KOMBAT to match them).
- **Manual alignment**: this method is unique to the RGB composition tool. When selected, the spin buttons change function from displaying information about alignment shifts to allowing you to enter desired shifts in x and y as well as rotation. Rotation defaults to centering on the image center, but a custom rotation center may be chosen for each layer using the centermark button at the right of each set of layer controls. This method may be of use if you have imagery that is not suited to alignment using any of the other methods, or if you wish to fine tune results.

You can choose the framing method that will be applied to your aligned layers. The following methods are possible:

- **First channel** - this method frames all other frames to the first channel (usually Luminance or Red). This makes particular sense for LRGB compositions.

- **MIN** - this method applies the minimum framing: the largest rectangle including only regions that are covered by all the input images will be framed.
- **COG** - diese Methode wählt einen optimalen Bildausschnitt für die Komposition auf der Grundlage des Schwerpunkts der ausgerichteten Bildpixel.

Bemerkung

Framing methods do not apply to DFT shift alignment. This method always frames based on the first layer. If this alignment method is selected, the chosen framing method will be ignored.

Tipp

On alignment the RGB composition tool aligns the loaded images in memory. That means that if you decide you aren't happy with the alignment and want to try a different method or a different framing, you have to reload all the images to get back to their original state before alignment. The *Reload all* button is provided for this purpose.

If you prefer to composite your images in a different way, for example using Pixel Math, you can still use the RGB composition tool as a quick means of aligning the images. After alignment, pressing *Save all* will save all the aligned images with their original filenames prefixed with *comp_*. These can then be brought into whatever tool you like and, unlike if you create a sequence with your stacks and align it in the normal way, it is obvious from the filename which image is which.

This tool is by nature graphical and is not exactly replicated in a Siril command, however there are commands that can be used to achieve part of the process:

- *Registrierung* kann eine Sequenz mit Ihren gestackten Bildern ausrichten,
- The *rgbcomp* command can compose a RGB image from 3 separate R, G and B images or 4 L, R, G, B images, however the images must be pre-aligned and it cannot assign mixed colors to the input images.
- *Pixelmath* can also be used in scripts to combine channels.
- There is also a built-in RGB Composition script that can partly automate the process.

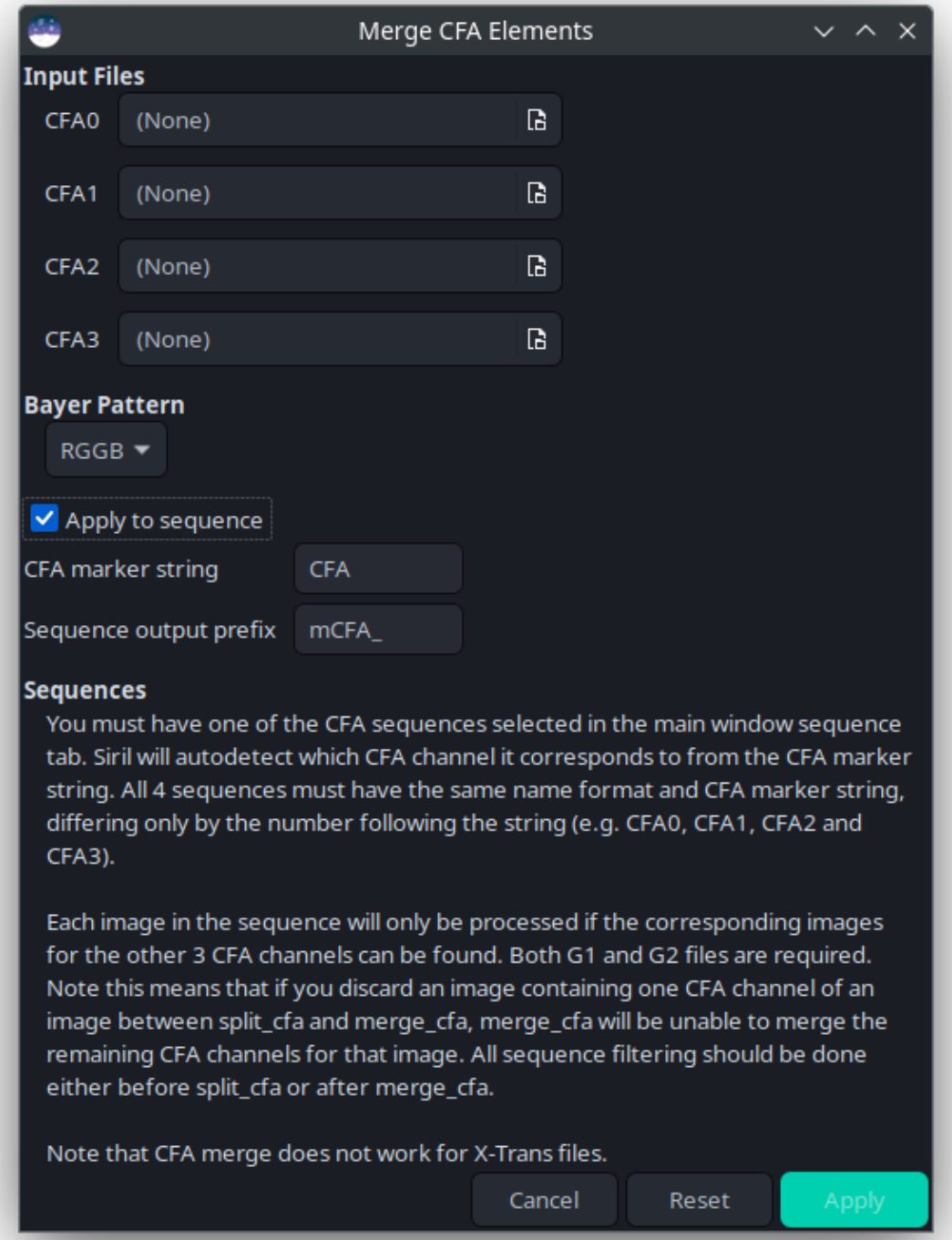
Siril Kommandozeile

```
rgbcomp red green blue [-out=result_filename] [-nosum]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename] [-nosum]
```

Erstellt eine RGB-Komposition aus drei unabhängigen Bildern oder eine LRGB-Komposition aus dem optionalen Luminanzbild und drei monochromen Bildern oder einem Farbbild. Das Ergebnisbild heißt composed_rgb.fit oder composed_lrgb.fit, es sei denn, es wird ein anderer Name im optionalen Argument angegeben. Ein weiteres optionales Argument, **-nosum**, weist Siril an, Belichtungszeiten nicht zu summieren. Dies wirkt sich auf FITS-Schlüsselwörter wie LIVETIME und STACKCNT aus

10.12 CFA Kanäle zusammenführen

Der Zweck dieses Werkzeugs besteht darin, mehrere monochrome Bilder zu kombinieren, die zuvor aus einem CFA-Sensor extrahiert wurden (z. B. mit dem Menü *Extraktion → CFA aufteilen...*). Das Werkzeug fügt die getrennten Bilder des roten, grünen (x2) und blauen Kanals zu einem einzigen zusammengesetzten Bild, dem CFA-Bild, zusammen.


Merge CFA Elements
Input Files
CFA0
(None)
CFA1
(None)
CFA2
(None)
CFA3
(None)
Bayer Pattern
RGGB
Apply to sequence
CFA marker string
CFA
Sequence output prefix
mCFA_
Sequences
You must have one of the CFA sequences selected in the main window sequence tab. Siril will autodetect which CFA channel it corresponds to from the CFA marker string. All 4 sequences must have the same name format and CFA marker string, differing only by the number following the string (e.g. CFA0, CFA1, CFA2 and CFA3).
Each image in the sequence will only be processed if the corresponding images for the other 3 CFA channels can be found. Both G1 and G2 files are required. Note this means that if you discard an image containing one CFA channel of an image between split_cfa and merge_cfa, merge_cfa will be unable to merge the remaining CFA channels for that image. All sequence filtering should be done either before split_cfa or after merge_cfa.
Note that CFA merge does not work for X-Trans files.
Cancel
Reset
Apply

Warnung

Dieses Tool ist für Bilder aus einer Bayer-Matrix gedacht und kann daher nicht mit Bildern aus X-Trans-Dateien von Fuji-Kameras arbeiten.

Der Dialog ist in drei verschiedene Teile gegliedert:

- **Eingabedateien**: Wählen Sie das Bild aus, das die Bayer-Teilmuster CFA0, CFA1, CFA2 und CFA3 enthält. Wenn dieses Bild mit der Siril-Funktion "Split CFA" erstellt wurde, hat es das Präfix "CFA".
- **Bayer-Muster**: Legt den Bayer-Muster-Header fest, der auf das Ergebnis angewendet werden soll. Dieses muss mit dem Bayer-Muster des Bildes übereinstimmen, aus dem die ursprünglichen Bayer-Subkanäle aufgeteilt wurden.
- Im unteren Teil des Dialogs können ganze Sequenzen verarbeitet werden, indem eine CFA-Bildsequenz rekonstruiert wird. Wenn Sie auf die Schaltfläche *Auf Sequenz anwenden* klicken, wird ein Hilfetext angezeigt, um korrekt vorzugehen. Dieser Text wird im nächsten Tooltip angezeigt. Es sind zwei Optionen verfügbar:
  - **Sequenz-Eingabemarkierung**: Kennzeichnungspräfix, das zur Bezeichnung der CFA-Nummer in den einzelnen CFA-Kanalbildern verwendet wird. Dies sollte auf das Sequenzpräfix gesetzt werden, das bei der Ausführung des split_cfa-Prozesses verwendet wurde (Standard: `CFA_`).
  - **Präfix der Sequenzausgabe**: Präfix der Bildnamen, die aus dem Merge-CFA-Prozess resultieren. Standardmäßig ist dies `mCFA_`.

Tipp

You may have any of the CFA0, CFA1, CFA2 or CFA3 sequences selected in the main window sequence tab: Siril will determine which one it is from the prefix and the number.

Your separate sub-CFA sequences should have been processed in exactly the same way; they **must** have the same sequence name format and CFA marker string, differing only but the number 0-3 following the CFA marker string (e.g. CFA0_bg_pp_lights, CFA1_bg_pp_lights, CFA2_bg_pp_lights and CFA3_bg_pp_lights).

Jedes Bild in der Sequenz wird nur verarbeitet, wenn die entsprechenden Bilder für die anderen 3 CFA-Kanäle gefunden werden können. Beide Bilder des grünen Subkanals sind erforderlich. Beachten Sie, dass dies bedeutet, dass merge_cfa die verbleibenden CFA-Kanäle für dieses Bild nicht zusammenführen kann, wenn Sie ein Bild verwerfen, das einen CFA-Kanal eines Bildes zwischen split_cfa und merge_cfa enthält. Die gesamte Sequenzfilterung sollte entweder vor split_cfa oder nach merge_cfa erfolgen.

Siril Kommandozeile

```
merge_cfa file_CFA0 file_CFA1 file_CFA2 file_CFA3 bayerpattern
```

Erzeugt ein Bayer-Farbbild aus 4 separaten Bildern, die die Daten der Bayer-Unterkanäle CFA0, CFA1, CFA2 und CFA3 enthalten. (Der entsprechende Befehl zur Aufteilung des CFA-Musters in Unterkanäle lautet **split_cfa**). Diese Funktion kann als Teil eines Arbeitsablaufs verwendet werden, bei dem die einzelnen Bayer-Farbkanäle vor dem Debayern verarbeitet werden. Der fünfte Parameter **bayerpattern** gibt das Bayer-Matrixmuster an, das neu erstellt werden soll: *bayerpattern* sollte einer der Werte 'RGGB', 'BGGR', 'GRBG' oder 'GBRG' sein

Siril Kommandozeile

```
seqmerge_cfa sequencename0 sequencename1 sequencename2 sequencename3 bayerpattern [-
↪prefixout=]
```

Fügt 4 Bildsequenzen zusammen, um das Bayer-Muster neu zu kombinieren. Die Sequenzen werden in den Argumenten **sequencename0**, **sequencename1**, **sequencename2** und **sequencename3** angegeben.

Das zu rekonstruierende Bayer-Muster muss als zweites Argument als eines der folgenden angegeben werden: RGGB, BGGR, GBRG oder GRBG (die Reihenfolge der Bayer-Kanäle muss mit der Reihenfolge der angegebenen Sequenzen übereinstimmen).

Hinweis: Alle 4 Eingabesequenzen **müssen** vorhanden sein und dieselben Abmessungen, Bittiefe und Bildanzahl haben.

Der Name der Ausgabesequenz beginnt mit dem Präfix "mCFA_" und einer Zahl, sofern mit der Option **-prefixout=** nichts anderes angegeben wurde

10.13 Pixel Math

Eines der leistungsfähigsten Werkzeuge in Siril ist die Pixelmathematik. Mit ihr können Sie die Pixel der Bilder mit mathematischen Funktionen manipulieren. Von einfacher Addition oder Subtraktion bis hin zu fortgeschrittenen Funktionen wie Modulationsübertragungsfunktion/MTF ist Pixel Math ein perfektes Werkzeug für die astronomische Bildverarbeitung.

Auf dieser Seite soll das Tool vollständig beschrieben werden. Detaillierte Beispiele finden Sie im ausgezeichneten Tutorial (englisch) auf der Website.

Das Fenster ist in 5 Bereiche unterteilt.

1. Die erste enthält 3 Textfelder, die die mathematischen Formeln enthalten. Nur daw erste Feld wird verwendet, wenn Sie ein monochromes Bild erzeugen wollen. Deaktivieren Sie die Schaltfläche *Nutze einzelnen RGB/K Ausdruck*, um eine RGB-Ausgabe zu erzeugen.
2. Der zweite ist der Variablenbereich mit der Auswahl von *Funktionen* und *Operatoren*. Jede Variable ist ein Bild, das zuvor mit der Schaltfläche + geladen werden muss. Sie können auf die gewünschte Funktion und/oder den gewünschten Operator klicken, um sie im Formelfeld erscheinen zu lassen.
3. Der dritte, der Bereich **Parameter**, erlaubt es dem Benutzer, Parameter zu definieren, die durch `,` getrennt sind. Wenn Sie zum Beispiel Parameter mit dem Ausdruck `Faktor=0,8, K=0,2` festlegen, werden alle Vorkommen von `Faktor` und `K` in der obigen Formel durch 0,8 bzw. 0,2 ersetzt. Die Formel `Ha * Faktor + OIII * K` würde also zu `Ha * 0,8 + OIII * 0,2` führen.
4. Das Feld **Ausgabe** ist für die Skalierung des Bildes innerhalb eines bestimmten Bereichs und für die Aktivierung oder Deaktivierung der Option *Belichtungszeit addieren* reserviert. Diese letzte Option gibt dem Benutzer die Möglichkeit, die Belichtungszeiten der einzelnen Bilder zu summieren oder nicht, und das Ergebnis im FITS-Header zu speichern. Vor der Verwendung muss der Rahmen erweitert werden.

   Pixel Math Skalierungsbox
5. Der Bereich **Voreinstellungen** schließlich ermöglicht es dem Benutzer, zuvor gespeicherte Formeln mit der Schaltfläche rechts neben den Formelbereichen wiederzuverwenden. Vor der Verwendung muss der Bereich aufgeklappt werden. Doppelklicken Sie auf die Formel, um sie in das korrekte Feld zu kopieren.

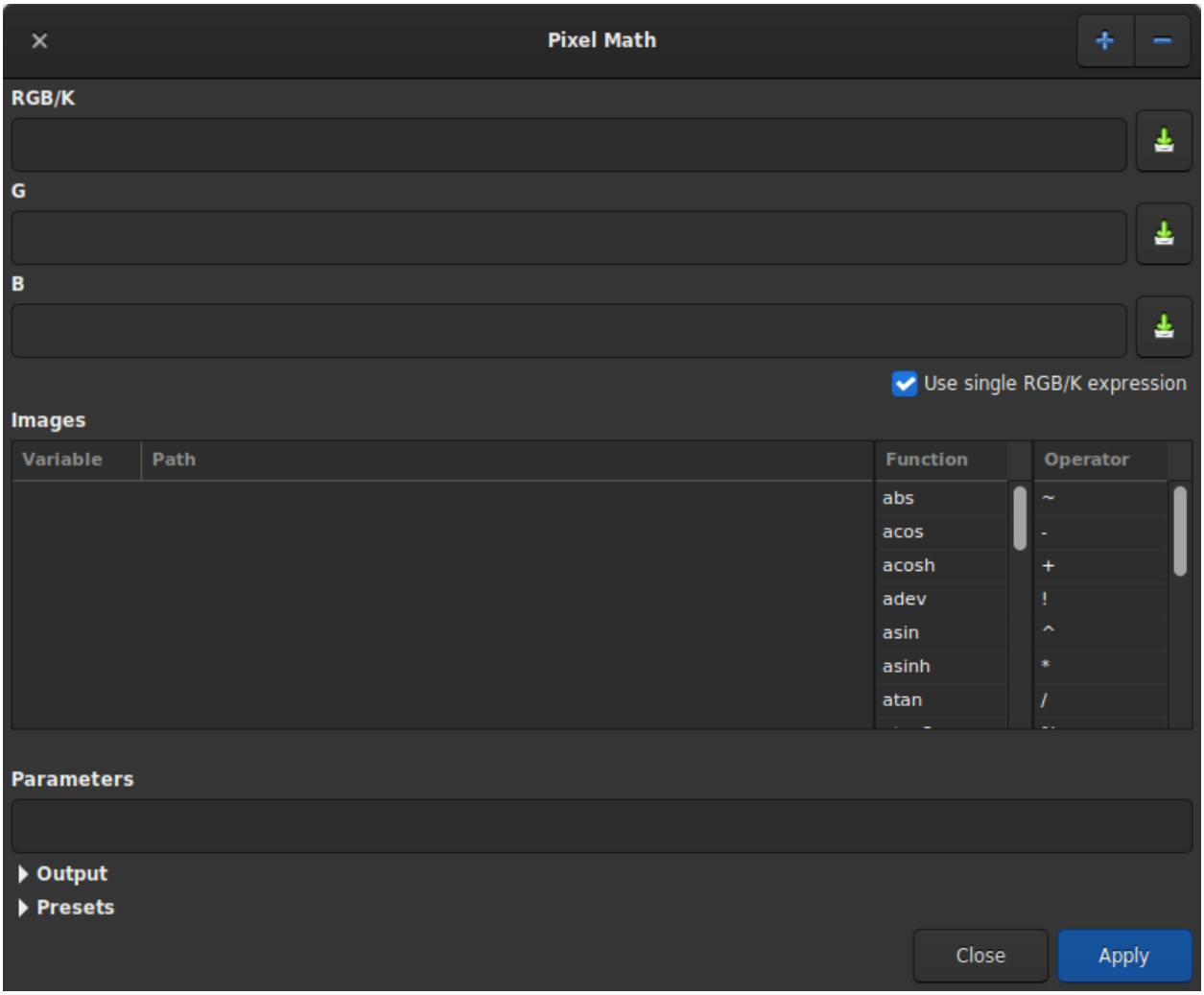


Abb. 87: Dialogfeld Pixel Math wie beim Öffnen gezeigt

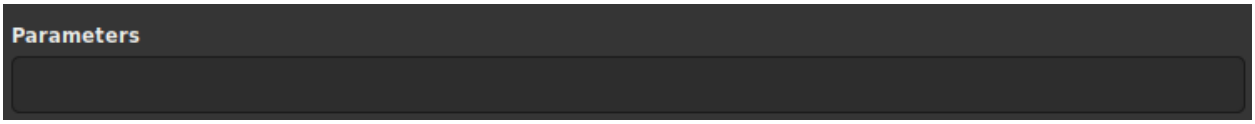


Abb. 88: Parameterbox Pixel Math

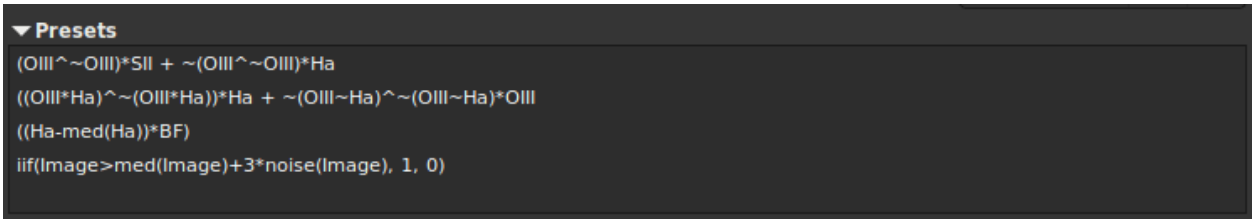


Abb. 89: Pixel Math Voreinstellungen

10.13.1 Benutzung

Namen der Variablen

Standardmäßig ist es möglich, 10 Bilder gleichzeitig zu laden. Jedem Bild wird ein Variablenname gegeben, der mit I beginnt, gefolgt von einer Zahl von 1 bis 10. Enthält das geladene Bild jedoch das Schlüsselwort `FILTER`, so wird der Wert dieses Schlüsselworts zum Standardvariablennamen. Natürlich ist es jederzeit möglich, ihn durch einen Doppelklick zu ändern.

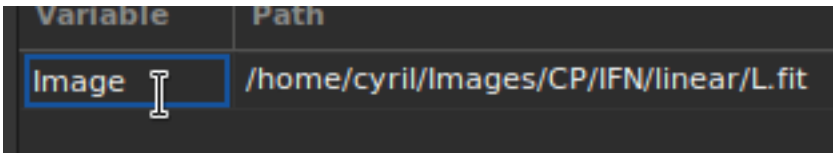


Abb. 90: Es ist möglich, den Namen der Variablen zu ändern.

`$T` is a reference to the current image, meaning the image on which the PixelMath operations will be applied.

Tipp

It is possible to use the currently loaded image by using the `$T` token. Note, however, that unlike other programs, the expression `$T[i]`, with `i=0,1,2`, is not recognized.

Beispiele

Nehmen wir ein monochromes Bild von Galaxien. Es handelt sich um lineare Daten, die durch die Autostretch-Ansicht betrachtet werden.

Der folgende Ausdruck:

```
iif(Image>med(Image)+3*noise(Image), 1, 0)
```

wird eine Sternmaske erzeugt.

Siril Kommandozeile

```
pm "expression" [-rescale [low] [high]] [-nosum]
```

Dieser Befehl wertet den im Argument angegebenen Ausdruck wie im PixelMath-Tool aus. Der vollständige Ausdruck muss in Anführungszeichen stehen, und die Variablen (d. h. die Bildnamen ohne Erweiterung, die sich in diesem Fall im Arbeitsverzeichnis befinden) müssen von dem Token $ umgeben sein, z. B. "$Bild1$ * 0,5 + $Bild2$ * 0,5". Es können maximal 10 Bilder in dem Ausdruck verwendet werden.

Abb. 91: Originalbild.

Abb. 92: Nach der obigen Formel.

Das Bild kann mit der Option **-rescale**, gefolgt von den Werten **low** und **high** im Bereich [0, 1], neu skaliert werden. Wenn keine niedrigen (low) und hohen (hight) Werte angegeben werden, werden die Standardwerte auf 0 und 1 gesetzt. Ein weiteres optionales Argument, **-nosum**, weist Siril an, Belichtungszeiten nicht zu summieren. Dies wirkt sich auf FITS-Schlüsselwörter wie LIVETIME und STACKCNT aus

10.13.2 Funktionen

Es gibt zwei Arten von Funktionen. Solche, die sich direkt auf die Pixel beziehen, und solche, die sich auf das gesamte Bild beziehen (z. B. die Statistikfunktionen).

Tab. 1: Pixelbasierte Funktionen

Funk-tion	Anwendung	Definition
abs	abs (x)	Absoluter Wert von x.
acos	acos (x)	Arkuscosinus von x.
acosh	acosh (x)	Hyperbolischer Arcuscosinus von x.
asin	asin (x)	Arcussinus von x.
asinh	asinh (x)	Hyperbolscher Arkussinus von x.
atan	atan (x)	Arkustangens von x.
atan2	atan2 (y, x)	Arcustangens von y/x.
atanh	atanh (x)	Arkustangens Hyperbolicus von x.
ceil	ceil (x)	Rundet x auf zur nächsten Ganzzahl.
cos	cos (x)	Cosinus von x.
cosh	cosh (x)	Cosinus Hyperbolicus von x.
e	e	Die Konstante e=2.718282...
exp	exp (x)	Exponentialfunktion.
fac	fac(x)	Fakultätsfunktion.
iif	iif(cond, expr_true, expr_false)	Bedingungsfunktion (oder einzelige Wenn-Funktion/inline if function). Liefert *expr_true*, falls *cond* zu einem Wert != 0 ausgewertet wird. Liefert *expr_false* falls *cond* zu 0 ausgewertet wird.
floor	floor (x)	Höchste ganze Zahl, die kleiner oder gleich x ist.
ln	ln (x)	Natürlicher Logarithmus von x.
log	log (x)	Logarithmus von x zur Basis 10.
log10	log10 (x)	Logarithmus von x zur Basis 10.
log2	log2 (x)	Logarithmus von x zur Basis 2.
max	max (x, y)	Maximum-Funktion. Liefert x, falls x>y, sonst y.
min	min (x, y)	Minimumfunktion.
mtf	mtf (m, x)	Modulationsübertragungsfunktion (MTF, Midtones Transfer Function) von x für einen Mitteltonausgleichsparameter m im Bereich [0, 1].
ncr	ncr (x, y)	Binominalkoeffizient "n choose r".
npr	npr (x, y)	Permutationsbefehl. Liefert die Anzahl der Permutationen ohne Wiederholungen von x Elementen aus einer Liste von y Elementen.
pi	pi	Die Konstante =3.141592...
pow	pow (x, y)	Exponentialfunktion x hoch y.

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Funktion	Anwendung	Definition
sign	sign (x)	Vorzeichen von x: $+1$ if $x > 0$ 1 if $x < 0$ 0 if $x = 0$.
sin	sin (x)	Sinus von x.
sinh	sinh (x)	Sinus Hyperbolicus von x.
sqrt	sqrt (x)	Quadratwurzel von x.
tan	tan (x)	Tangens von x.
tanh	tanh (x)	Tangens Hyperbolicus von x.
trunc	trunc (x)	Verkürzt x auf die vorherige Ganzzahl.

Tab. 2: Statistikfunktionen

Funktion	Anwendung	Definition
adev	adev (Image)	Durchschnittliche absolute Abweichung des Bildes.
bwmv	bwmv (Image)	Bi-gewichtete Mittelvarianz des Bildes (Biweight midvariance).
height	height (Image)	Höhe des angegebenen Bildes in Pixeln.
mad	mad (Image)	Mittlere absolute Abweichung des Bildes. mdev kann ebenfalls genutzt werden.
max	max (Image)	Pixel-Maximalwert des Bildes.
mean	mean (Image)	Mittelwert der Pixel des Bildes.
med	med (Image)	Median der Pixelwerte des Bildes. Es kann auch "median" verwendet werden.
min	min (Image)	Minimaler Pixelwert des Bildes.
noise	noise (Image)	Abschätzung des gaußschen Rauschens des Bildes.
sdev	sdev (Image)	Standardabweichung der Pixelwerte des Bildes.
width	width (Image)	Breite des Bildes in Pixeln.

10.13.3 Operatoren

Tab. 3: Operatoren

Operator	Anwendung	Definition
~	~x	Pixel-Inversoperator.
-	-x	Unärer Minus-Operator (Vorzeichenwechsel).
+	+x	Unärer Plus-Operator.
!	!x	Logischer NICHT-Operator.
^	x ^ y	Exponentialoperator.
*	x * y	Multiplikationsoperator.
/	x / y	Divisionsoperator.
%	x % y	Modulooperator.
+	x + y	Additionsoperator.
-	x - y	Subtraktionsoperator.
<	x < y	Kleiner als Vergleichsoperator.
<=	x <= y	Kleiner-Gleich relationaler Operator.
>	x > y	Größer als Vergleichsoperator.
>=	x >= y	Größer-Gleich relationaler Operator.
==	x == y	Gleichheitsoperator.
!=	x != y	Relationaler Ungleichheits-Operator.
&&	x && y	Logscher Operator UND.
||	x || y	Logischer Operator ODER.

KAPITEL 11

Astrometrie

Astrometrie ist die Wissenschaft, die sich mit der Position und der Bewegung von Himmelskörpern beschäftigt. Astrometrie ist in der modernen Astronomie unverzichtbar, wo Aufnahmesoftware wie N.I.N.A., Ekos, APT oder Andere die Bilder vermessen um die Lage eines Bildes im Bezug zum Himmel genau zu bestimmen (Plate Solve). Die Astrometrie kann auch in der Verarbeitungsphase verwendet werden, z.B. bei der photometrischen Farbkalibrierung.

11.1 Platesolving/astrometrische Lösung

Die Plattenauflösung ist ein wichtiger Schritt in der astronomischen Bildverarbeitung. Sie ermöglicht es, Bilder mit Himmelskoordinaten zu verknüpfen, so dass man weiß, welches Objekt sich im beobachteten Sichtfeld befindet. Viele der Siril-Werkzeuge, wie z. B. die spektrophotometrische oder photometrische Farbkalibrierung (SPCC oder PCC), müssen die Koordinaten des Bildes mit ausreichender Genauigkeit kennen, damit sie funktionieren.

Die Astrometrie in Siril kann mit zwei verschiedenen Werkzeugen durchgeführt werden :

- Mit dem speziellen Werkzeug, das über das Menü *Werkzeuge* → *Astrometrie* → *Photometrische Lösung*, oder über die Tastenkombination `Strg` + `Umschalttaste` + `A` zugänglich ist.
- Durch Benutzung des Befehls `platesolve`, eingeführt in Siril 1.2.

Siril Kommandozeile

```
platesolve [-force] [image_center_coords] [-focal=] [-pixelsize=]
platesolve ... [-noflip] [-downscale] [-order=] [-radius=] [-disto=]
platesolve ... [-limitmag=[+-]] [-catalog=] [-nocrop]
platesolve ... [-localasnet [-blindpos] [-blindres]]
```

Berechnet eine astrometrische Lösung für das geladene Bild.

Wenn das Bild bereits eine astrometrische Lösung besitzt, wird nichts getan, es sei denn, das Argument **-force** wird übergeben, um eine neue Lösung zu erzwingen. Wenn WCS oder andere Bild-Metadaten fehlerhaft sind oder fehlen, müssen Argumente übergeben werden:

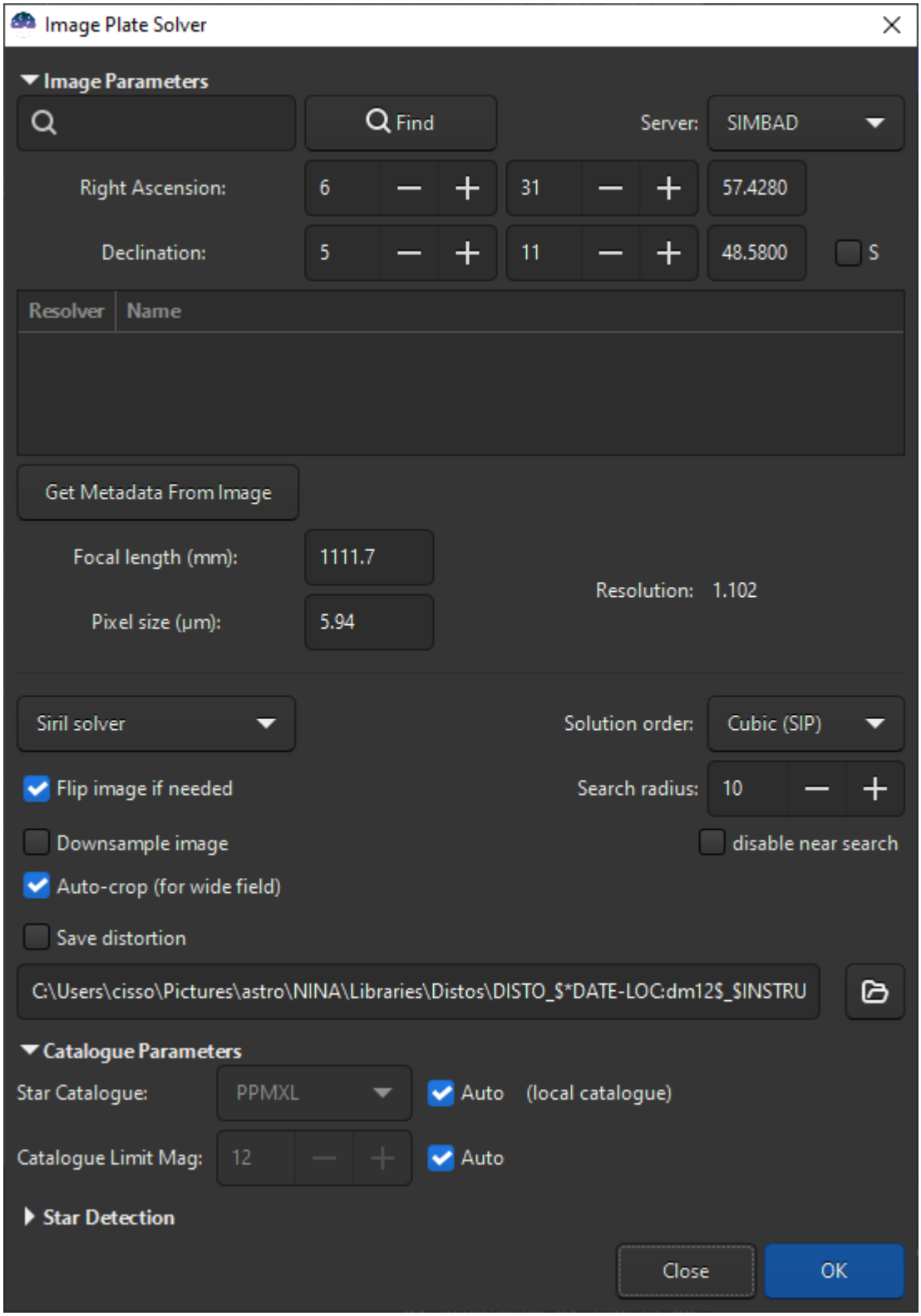


Abb. 1: Dialog Astrometrische Lösung

Die ungefähren Koordinaten des Bildmittelpunkts können in Dezimalgraden oder Grad/Stunde-Minute-Sekunde-Werten (J2000 mit Doppelpunkten) angegeben werden, wobei die Werte für Rektaszension und Deklination durch ein Komma oder ein Leerzeichen getrennt sind (für astrometry.net nicht obligatorisch).

Brennweite und Pixelgröße können mit **-focal=** (in mm) und **-pixelsize=** in (in µm) übergeben werden, wobei die Werte aus dem Bild bzw. den Einstellungen überschrieben werden. Siehe auch unter Optionen zum blinden Lösen mit lokalem Astrometry.net

Wenn **-noflip** nicht angegeben ist, wird das Bild gespiegelt, wenn es als auf dem Kopf stehend erkannt wird.

Für eine schnellere Erkennung von Sternen in großen Bildern ist eine Verkleinerung des Bildes mit **-downscale** möglich.

Bei der Lösung können Verzerrungen nach der SIP-Konvention mit Polynomen bis zur Ordnung 5 berücksichtigt werden. Standardmäßig wird die kubische Verzerrung aus den Astrometrie-Einstellungen genutzt. Dies kann mit der Option **-order=** geändert werden, die einen Wert zwischen 1 und 5 angibt.

Wenn Sie lokale Siril-Solver-Kataloge oder lokales Astrometry.net verwenden und die anfängliche Lösung nicht erfolgreich ist, sucht der Solver nach einer Lösung innerhalb eines Radiuskegels, der mit der Option **-radius=** angegeben wird. Wenn kein Wert übergeben wird, wird der Suchradius aus den Astrometrieeinstellungen übernommen. Die Siril-Nähesuche kann durch Übergeben eines Werts von 0 deaktiviert werden. (kann für Astrometry.net nicht deaktiviert werden).

Mit der Option **-disto=** können Sie die aktuelle astrometrische Lösung als Verzerrungsdatei speichern.

Bilder können entweder mit Siril unter Verwendung eines Sternkatalogs und des globalen Registrierungsalgorithmus oder mit dem lokalen solve-field-Befehl von astrometry.net (aktiviert mit **-localasnet**) astrometrisch gelöst werden.

Optionen des Siril Lösers:

Die Grenzgröße der Sterne, die für die Plattenlösung verwendet werden, wird automatisch aus der Größe des Sichtfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.

Die Wahl des Sternkatalogs erfolgt automatisch, es sei denn, die Option **-catalog=** wird übergeben: Wenn lokale Kataloge installiert sind, werden sie verwendet, andernfalls erfolgt die Wahl auf der Grundlage des Sichtfelds und der Grenzgröße. Wenn die Option übergeben wird, erzwingt sie die Verwendung des im Argument angegebenen Katalogs, mit den möglichen Werten: tycho2, nomad, localgaia, gaia, ppmxl, brightstars, apass.

Wenn das berechnete Bildfeld größer als 5 Grad ist, wird die Sternerkennung auf eine beschnittenen Bereich um das Bildzentrum begrenzt, falls nicht die Option **-nocrop** verwendet wird.

Optionen des Astrometry.net Lösers:

Die Übergabe der Optionen **-blindpos** und/oder **-blindres** ermöglicht die blinde Lösung nach Position bzw. nach Sampling. Sie können diese verwenden, wenn Sie ein Bild mit einem völlig unbekannten Bildausschnitt und einem unbekannten Sampling lösen

Seit Version 1.2 kann die astrometrische Lösung mit zwei verschiedenen Algorithmen durchgeführt werden. Der erste war bis zu dieser Version der einzige in Siril, er basiert auf der globalen Sternregistrierung und versucht, Bilder auf ein virtuelles Bild eines Katalogs mit demselben Gesichtsfeld zu registrieren. Das zweite ist neu, es benutzt ein externes Programm namens `solve-field` aus der Astrometry.net Suite, das lokal installiert wird. Für Windows-Plattformen ist der einfachste Weg, es zu bekommen, ansvr zu benutzen.

Seit Version 1.3 sucht der interne Siril-Löser auch innerhalb eines Kegels um die ursprünglichen Zielkoordinaten, wenn keine Übereinstimmung gefunden wurde. Dies ist nur verfügbar, wenn *lokale Sternkataloge* verwendet werden. Das Protokoll zeigt in diesem Fall folgende Informationen an:

```
Initial solve failed
Attempting a near solve with a radius of 10.0 degrees
```

Für astrometrische Lösungen müssen einige Parameter ermittelt werden, z. B. die Abtastrate. Das Fenster des Tools hilft bei der Erfassung dieser Parameter. Wir werden nun sehen, wie man sie korrekt ausfüllt.

11.1.1 Bildparameter

Zielkoordinaten

Die Suche nach einer astrometrischen Lösung ist einfacher und schneller, wenn wir ungefähr wissen, wo wir suchen. Sirils Plate-Solver muss, da er einen Katalog mit dem Bild vergleicht, die ungefähre Position des Bildmittelpunkts kennen, um den richtigen Katalogauszug zu erhalten. Astrometry.net hat alle Kataloge, die es braucht lokal gespeichert, so dass sie alle durchsucht werden können, um eine Lösung zu finden, aber es ist natürlich viel schneller, anzugeben anfangen werden soll.

Die Aufnahmesoftware steuert heutzutage oft auch das Teleskop und sollte die ungefähren Koordinaten kennen, an denen das Bild aufgenommen wurde. In diesem Fall werden diese Koordinaten bei Verwendung des FITS-Formats in den Metadaten des Bildes, dem FITS-Header, angegeben. Das ist nicht immer der Fall, und schon gar nicht, wenn RAW-DSLR-Bilder anstelle von FITS erstellt werden.

Beim Öffnen des Plate Solver-Fensters werden die Metadaten des aktuellen Bildes geladen und im Fenster angezeigt. Wenn oben keine Koordinaten angezeigt werden oder wenn RA und Dec auf Null bleiben, sind einige Benutzereingaben erforderlich. Wenn Sie überhaupt nicht wissen, um welches Bild es sich handelt, verwenden Sie eine Blindlösung mit astrometry.net. Andernfalls geben Sie die äquatorialen J2000-Koordinaten an, die so nah wie möglich am Bildzentrum liegen, indem Sie entweder die Felder ausfüllen, wenn Sie die Koordinaten bereits kennen, oder eine Abfrage mit einem Objektnamen durchführen (noch nicht über den Befehl möglich).

Das Textfeld oben links im Fenster ist das Suchfeld, dessen Ziel es ist, einen Objektnamen in seine Koordinaten umzuwandeln. Wenn Sie `ENTER` drücken oder auf die Schaltfläche *Suchen* klicken, wird das Objekt zunächst in den lokalen Beschriftungskatalogen gesucht. Wenn es nicht gefunden wird, wird eine Web-Anfrage gestellt, um seine Koordinaten zu erhalten. Es können mehrere Ergebnisse für den eingegebenen Namen gefunden werden, die in der Liste unten angezeigt werden. Durch Auswahl eines Ergebnisses werden die Koordinaten am oberen Rand aktualisiert, standardmäßig ist das erste Ergebnis ausgewählt.

Es ist auch möglich, den Server auszuwählen, auf dem die Abfrage ausgeführt werden soll. Dies ändert die Ergebnisse nicht wesentlich, aber manchmal kann einer der Server offline sein, so dass andere als Backup fungieren. Auswahl zwischen CDS, VizieR und SIMBAD (Standard).

Bemerkung

Wenn das Objekt nicht gefunden wird, versuchen Sie es bitte mit dem vollständigen Namen oder mit dem Namen aus einem Katalog. Die Beschriftungskataloge enthalten einige gebräuchliche Namen, die Online-Dienste auch, aber nicht alle, und sie finden keine Teilantworten. Für den Blasennebel zum Beispiel geben Sie bitte `NGC 7635` oder `bubble nebula` ein, nicht `bubble`. Die Katalogbezeichnungen der meisten Kataloge entstammen der englischen Sprache. Also `horsehead` und nicht `Pferdekopf`.

Die Koordinatenfelder werden automatisch ausgefüllt, aber Sie können auch Ihre eigenen Koordinaten eingeben. Vergessen Sie nicht, das Feld "S" anzukreuzen, wenn sich das gesuchte Objekt auf der Südhalbkugel des Himmels befindet (negative Deklinationen).

Abtastrate

Die Abtastrate ist der wichtigste Parameter für die Ermittlung der astrometrischen Lösung. Sie wird in Bogensekunden pro Pixel angegeben und gibt an, wie stark das Bild auf den Himmel gezoomt ist, d. h. wie breit das zu suchende Feld ist.

Sie wird von zwei Parametern abgeleitet: Brennweite und Pixelgröße. Sie sind häufig auch in den Metadaten des Bildes verfügbar. Wenn sie nicht aus der Bilddatei verfügbar sind, werden die in den Einstellungen gespeicherten Werte verwendet. Die Werte der Bilder und der Einstellungen können über den Dialog *Information* eingestellt werden. Überprüfen Sie in jedem Fall den angezeigten Wert vor der astrometrischen Lösung und korrigieren Sie ihn gegebenenfalls. Wenn eine astrometrische Lösung gefunden wird, werden die Standardwerte für Brennweite und Pixelgröße überschrieben. Dieses Verhalten kann in den Einstellungen deaktiviert werden.

Warnung

Wenn Binning verwendet wurde, sollte es im FITS-Header angegeben sein, aber dies kann auf zwei Arten geschehen: Die Pixelgröße kann gleich bleiben und der Binning-Multiplikator sollte zur Berechnung des Samplings verwendet werden, oder die Pixelgröße wird bereits von der Erfassungssoftware multipliziert. Je nach Fall kann eine der beiden Formen in den Voreinstellungen oder im Fenster *Informationen* gewählt werden.

Die Pixelgröße ist in den technischen Daten der astronomischen Kameras angegeben und kann im Allgemeinen im Internet für DSLR- oder andere Kameras gefunden werden. Es gibt nur eine begrenzte Anzahl verwendeter Bildsensoren und die Daten der meisten Sensoren sind bekannt.

Die Brennweite hängt vom Hauptinstrument, aber auch vom Backfocus und den verwendeten Korrektur- oder Zoomobjektiven ab. Geben Sie einen Wert an, der Ihrer Meinung nach der effektiven Brennweite am nächsten kommt. Wenn eine astrometrische Lösung gefunden wird, wird die berechnete Brennweite in den Ergebnissen gespeichert, und Sie können diese in Ihrer Aufnahmesoftware und für künftige Verwendungen des Tools wiederverwenden.

Wenn eines der Felder aktualisiert wird, wird das Sampling neu berechnet und im Fenster angezeigt (hier "Auflösung" genannt). Achten Sie darauf, dass der Wert so nah wie möglich an der Realität ist.

Tipp

In der GUI in oranger Farbe geschriebene Daten weisen auf Werte hin, die nicht aus dem Bildheader abgerufen werden konnten. Dies bedeutet nicht, dass sie falsch (sie könnten aus den Einstellungswerten geladen worden sein und gültig sein) oder obligatorisch (Sie könnten blind lösen) sind. Diese Farbe dient lediglich zur Unterscheidung von aus der Kopfzeile gelesenen Werten.

11.1.2 Parameter des Lösers

Seit Siril 1.2 kann Siril zwei verschiedene Löser verwenden, seinen internen Löser und *Astrometry.net lokale Installation*. Die Schnittstelle unterscheidet sich je nachdem, ob das eine oder das andere in der entsprechenden Dropdown-Liste ausgewählt wird.

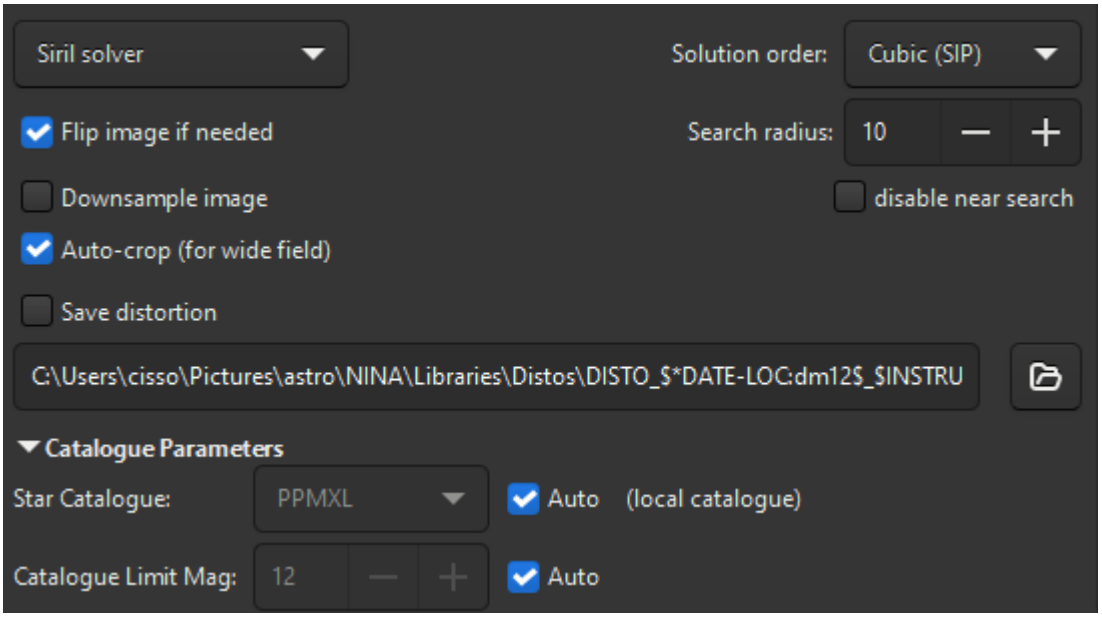


Abb. 2: Optionen des internen Siril-Lösers

Verzerrungen

Die Option *Lösungsreihenfolge*, die über die Dropdown-Liste ausgewählt wird, gibt die Reihenfolge der astrometrischen Lösung an. Wenn Sie `Linear` wählen, versucht der Algorithmus eine Lösung zu finden, die davon ausgeht, dass keine Verzerrungen im Bild vorhanden sind (d.h. das Feld ist optisch flach). Diese Annahme kann jedoch bei Vorhandensein von optischen Abbildungsfehlern (falscher Backfocus, kein Field-Flattener etc...) unzutreffend sein. Seit Version 1.3 kann der Plattelöser versuchen, polynomiale Verzerrungen bis zur fünften Ordnung anzupassen, wobei er der SIP-Konvention <https://irsa.ipac.caltech.edu/data/SPITZER/docs/files/spitzer/shupeADASS.pdf>`_ folgt. Standardmäßig verwendet der Platesolver `kubische (SIP)` Polynome, die für die meisten Anwendungsfälle geeignet sein sollten. Diese Standardeinstellung kann in den *Einstellungen* geändert werden. Diese Option ist bei beiden Lösern verfügbar.

Sie können Verzerrungen auch in einer WCS-Datei speichern, indem Sie das Kontrollkästchen *Verzerrung speichern* aktivieren. Verwenden Sie den Dateiwähler unten, um den Pfad anzugeben. Wenn Sie in den *Einstellungen* einen Hauptverzerrungspfad definiert haben, wird sein Pfad standardmäßig angezeigt. Diese Datei kann dann verwendet werden, um Bilder während der *Globalen Registrierung* und 2pass-Registrierung zu entzerren.

Suchradius

Wenn *lokale Kataloge* installiert sind oder wenn *Astrometry.net* verwendet wird, wird ein Kegel um die Zielposition durchsucht. Die Größe dieses Kegels in Grad kann mit dem *Suchradius*-Steuerelement geändert werden, dessen Standardwert in den *Einstellungen* geändert werden kann. Für den Siril-Löser kann diese Funktion durch Ankreuzen des Kontrollkästchens „Nahbereichs-Suche deaktivieren“ deaktiviert werden.

Blinder Löser Astrometry.net

Wenn der Astrometry.net-Löser ausgewählt ist, stehen zwei zusätzliche Optionen zur Verfügung:

1. *Position ignorieren (blinde Lösung)* ermöglicht das Ignorieren der in der Zielkoordinatensteuerung angegebenen Position.
2. *Sampling ignorieren (blinde Lösung)* ermöglicht das Ignorieren des aus der Pixelgröße und Brennweite berechneten Samplings.

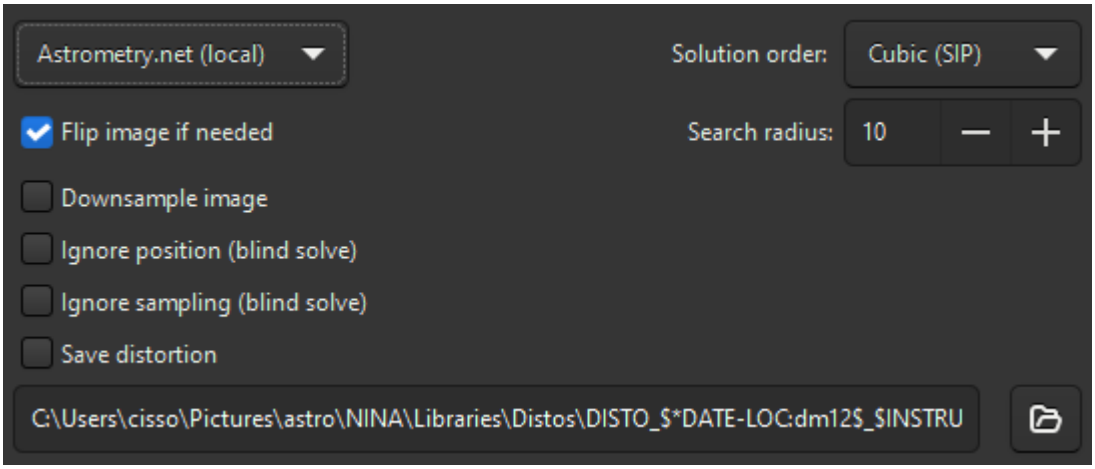


Abb. 3: Optionen des Astrometry.net Lösers

Verwenden Sie diese beiden Optionen zusammen, wenn das Ziel und das Sampling des Bildes völlig unbekannt sind.

> **Warnung**
>
> Hier passiert keine Magie. Um eine Lösung für ein beliebiges Sichtfeld zu finden, müssen die erforderlichen *Indizes* installiert sein, die das tatsächliche Sichtfeld des zu lösenden Bildes abdecken.

Andere Parameter

Schließlich gibt es noch drei Schaltflächen am unteren Rand des Dialogs, die Sie umschalten können:

1. `Bilder herunterrechnen`: Diese Option führt ein Downsampling des Eingabebildes durch, bevor versucht wird, es zu lösen. Dies ist in einigen Fällen für gedrizzelte Bilder nützlich und beschleunigt den Prozess. Die Größe des Ausgangsbildes bleibt dabei unverändert.
2. Wenn das Bild von der astrometrischen Lösung als auf dem Kopf stehend erkannt wird und die Option *Bild bei Bedarf spiegeln* aktiviert ist, wird es am Ende gespiegelt. Dies kann je nach Aufnahmesoftware nützlich sein, wenn das Bild nicht die richtige Ausrichtung hat, wenn es in Siril angezeigt wird (siehe mehr *Erläuterungen*).
3. `Automatischer Crop (für Weitfeld)`: Wenn diese Option aktiviert ist, wird nur in der Mitte des Bildes eine eine astrometrische Lösung durchgeführt. Dies macht man nur bei Bildern mit einem großem Bildfeld (größer als 5 Grad), bei denen Verzerrungen außerhalb des Zentrums groß genug sind, um das Werkzeug zu behindern. Wird bei Verwendung des astrometry.net-Solvers ignoriert.

11.1.3 Katalogparameter

This section is relevant for Siril internal solver only. Several online catalogues can be used and also two catalogues that can be installed locally for faster and more reliable operation.

Online-Kataloge verwenden

Standardmäßig ist dieser Bereich inaktiv, da alles auf automatisch eingestellt ist. Wenn Sie das Kästchen "Automatisch" deaktivieren, können Sie den Online-Katalog auswählen, der für die astrometrische Lösung verwendet wird, was von der Auflösung des Bildes abhängen kann. Die Auswahl erfolgt zwischen:

- TYCHO2, ein Katalog mit Positionen, Eigenbewegungen und zweifarbigen photometrischen Daten für 2.539.913 der hellsten Sterne in der Milchstraße.
- NOMAD <https://cdsarc.cds.unistra.fr/viz-bin/cat/I/297>`_, eine Zusammenführung von Daten aus den Katalogen Hipparcos, Tycho-2, UCAC2, Yellow-Blue 6 und USNO-B für Astrometrie und optische Photometrie, ergänzt durch 2MASS Nahinfrarot. Der fast 100 GB große Datensatz enthält astrometrische und photometrische Daten für etwa 1,1 Milliarden Sterne.
- Gaia DR3 <https://cdsarc.cds.unistra.fr/viz-bin/cat/I/355>`_, veröffentlicht am 13. Juni 2022. Die fünfparameterige astrometrische Lösung, Positionen am Himmel (,), Parallaxen und Eigenbewegungen werden für rund 1,46 Milliarden Quellen mit einer Grenzgröße von G = 21 angegeben. Dies entspricht dem neuesten Stand der Technik im Bereich der präzisen Astrometriedaten und liefert Spektraldaten für eine Vielzahl von Quellen, die in SPCC verwendet werden.
- PPMXL, ein Katalog mit Positionen, Eigenbewegungen, 2MASS- und optischer Photometrie von 900 Millionen Sternen und Galaxien.
- Bright Stars <https://cdsarc.cds.unistra.fr/viz-bin/cat/V/50>`_, ein Sternkatalog, der alle Sterne mit einer Sterngröße von 6,5 oder heller auflistet, d. h. ungefähr alle von der Erde aus mit bloßem Auge sichtbaren Sterne. Der Katalog enthält 9.110 Objekte.
- APASS <https://cdsarc.cds.unistra.fr/viz-bin/cat/II/336>`_, ein Sternkatalog, der alle Sterne mit einer Sterngröße von 6,5 oder heller auflistet, d. h. ungefähr alle von der Erde aus mit bloßem Auge sichtbaren Sterne. Der Katalog enthält 9.110 Objekte.

> **Bemerkung**
>
> Für die Nutzung dieser Online-Kataloge ist eine Internetverbindung erforderlich.

Alle diese Kataloge sind über das VizieR-Katalogzugriffstool, CDS, Straßburg, Frankreich, verfügbar (DOI:10.26093/cds/vizier). Die ursprüngliche Beschreibung des VizieR-Dienstes wurde im Jahr 2000 veröffentlicht, A&AS 143, 23.

Die Option `Katalog Mag-Limit` ist eine Option, mit der Sie die Helligkeit der im Katalog gefundenen Sterne begrenzen können. Der automatische Wert wird anhand der Bildauflösung berechnet.

Verwendung der lokalen KStars-Kataloge (NOMAD)

Mit der Version 1.1, die im Juni 2022 erschien, war es möglich, einen lokal installierten Sternkatalog zu verwenden, um den Funktion ohne Internetverbindung und mit größerer Ausfallsicherheit zu gewährleisten. Der Sternkatalog, den wir für unsere Bedürfnisse am besten geeignet fanden, ist der von KStars. Er besteht in der Tat aus vier Katalogen (dokumentiert hier in KStars), von denen zwei nicht direkt in den KStars-Installationsdateien enthalten sind:

- **namedstars.dat**, die hellsten Sterne, die alle einen Namen haben
- **unnamedstars.dat**, ebenfalls helle Sterne, aber bis zur Größenklasse 8

- **deepstars.dat**, schwächere Sterne aus dem Tycho-2-Katalog der 2,5 Millionen hellsten Sterne, bis zur Helligkeit 12,5
- **USNO-NOMAD-1e8.dat**, ein Auszug aus dem riesigen NOMAD-Katalog, der sich auf photometrische B-V-Informationen und die Eigenbewegung von Sternen in kompakter Form bis zur Helligkeit 18 beschränkt.

Dieser Katalog kann für die Plattenauflösung und für PCC verwendet werden.

Beim Vergleich dieser Kataloge mit dem Online-Katalog NOMAD kann man leicht feststellen, dass viele Sterne fehlen. Wenn nicht genügend Sterne für Ihr Bildfeld gefunden werden, sollten Sie trotzdem die Remoteabfragen verwenden. Wenn die Kataloge installiert sind, kann man mit dem Befehl *conesearch* überprüfen, welche Sterne des Bildes für die PCC verwendet werden, nämlich diejenigen, die mit photometrischen Informationen in den Katalogen verfügbar sind.

```
conesearch -phot
```

Verwendung der lokalen Gaia DR3-Kataloge

In version 1.4 support for a local extract of the Gaia DR3 catalogue has been added. This *astrometric* Gaia extract differs from the other catalogues in that it is focused on astrometry as applied to plate solving, so instead of defining a limiting magnitude the catalogue has been designed to provide an even coverage of stars in each level 8 HEALPixel (you can read more about HEALPixels here). This avoids including excessive numbers of stars in densely populated parts of the sky and ensures that sufficient fainter stars are included in emptier HEALPixels to support accurate plate solving.

The default catalogue will be selected at startup based on two criteria:

- If a local catalogue is available it is preferred to the remote equivalent
- If a Gaia catalogue (local or remote) is available it is preferred to NOMAD

> **Tipp**
>
> Although the offline Gaia DR3 extract can be used with the *conesearch* command (by specifying the catalogue name *localgaia*) it is not really designed for this purpose. In busy regions of the sky the limiting magnitude will be relatively bright and therefore many stars visible in your images will not be contained in the catalogue whatever limiting magnitude you set. For annotation cone searches in such regions you will be better served by specifying the catalogue name *gaia* instead, which will conduct a remote cone search using the Vizier mirror of the Gaia DR3 archive. This is both fast (for a remote query) and comprehensive.

This catalogue can be used only for plate solving. A *photometric* Gaia extract also exists, and can be used for the SPCC, it is documented on the *SPCC page*.

Download KStars NOMAD

Die ersten beiden Dateien sind im KStars-Quelltext verfügbar, der Tycho-2-Katalog aus einem Debian-Paket und der NOMAD-Katalog ebenfalls aus KStars-Dateien, wie in diesem kleinen Artikel zur KStars-Installation dokumentiert. Es gibt mehrere weltweite Spiegel wie in den Artikeln angegeben.

Um es den Siril-Benutzern und möglicherweise auch den KStars-Benutzern leichter zu machen, stellen wir die vier Dateien an einem einzigen Ort und in einem stärker komprimierten Format zur Verfügung. Mit dem LZMA-Algorithmus (verwendet von xz oder 7zip) beträgt die Dateigröße 1,0 GB anstelle der 1,4 GB der ursprünglichen gzip-Datei. Da diese *.xz*-Dateien komprimiert sind, müssen Sie sie vor dem Kopieren mit einer geeigneten Software dekomprimieren.

Direkte Download-Links finden Sie hier <https://free-astro.org/download/kstars-siril-catalogues/>`__ (klicken Sie mit der rechten Maustaste auf jeden Dateinamen auf der linken Seite und speichern Sie die Links).

Falls der Link nicht verfügbar ist und es schneller von überall her verfügbar zu machen, wird es mit bittorrent verteilt, torrent-Datei oder über den folgenden magnet link.

Download Astrometric Gaia DR3

The Gaia DR3 local astrometric extract is available from Zenodo It comes with a sha256 checksum for the compressed archive.

Full details, plus a citation reference and a checksum for the uncompressed archive for extra paranoia, can be found at the Zenodo record. The specification of the Gaia DR3 catalogue extracts and their file format is documented here (PDF).

> **Tipp**
>
> For convenience, an offline version of the Gaia DR3 astrometric catalogue can be installed using the Catalogue Installer script in the *Scripts* menu. This catalogue takes up 1.5GB once installed compared with 2.1GB for the KStars NOMAD catalogue and provides higher precision astrometry and even density of stars in all regions, for more reliable and accurate plate solving. **It is strongly recommended to install and use this catalogue for all astrometry purposes within Siril**.

Installation in Siril

Die Dateien können an beliebiger Stelle abgelegt und ihre Pfade in den Einstellungen von Siril angegeben werden, aber es gibt einen Standardspeicherort für die vier Dateien: `~/.local/share/kstars/` unter Linux. Sie können dort verlinkt werden, um unnötige Kopien zu vermeiden. Die Einstellungen können nun von der Kommandozeile aus geändert werden, indem man den Befehl :ref:`set <set>`verwendet.

Wenn verfügbar und lesbar, verwendet Siril den Webdienst nicht, um astrometrische oder photometrische Daten abzurufen. Überprüfen Sie anhand der Meldungen auf der Registerkarte Protokoll oder auf der Konsole, ob die Katalogdateien wie erwartet verwendet werden.

Ab Siril 1.4.0 sucht Siril zuerst in *Lokalen Beschriftungskataloge*, um die Koordinaten eines Objekts zu finden, das im Dialog der astrometrischen Lösung übergeben wurde, um den Mittelpunkt des Bildes zu lokalisieren. Dies bedeutet, dass Sie, sofern Sie die lokalen Sternkataloge installiert haben, Ihre Bilder ohne Internetverbindung lösen können. Dies ist natürlich nur erforderlich, wenn die Aufnahmesoftware die Zielkoordinaten nicht im FITS-Header gespeichert hat oder wenn Sie das SER-Dateiformat verwenden, das diese Informationen nicht speichern kann.

Benutzung

Mit dem Hinzufügen der neuen Verbindung zwischen Sirils Plattenlöser und dem lokalen Katalog und der neuen Verbindung zwischen Sirils PCC und dem lokalen Katalog wurde ein neuer Befehl `conesearch` geschaffen (ab Siril 1.4.0), um Katalogobjekte in einem astrometrisch gelösten Bild anzuzeigen. Um Sterne anzuzeigen, die photometrische Informationen (den B-V-Index) enthalten und für die Kalibrierung verwendet werden können, können Sie zum Beispiel folgendes verwenden:

```
conesearch -phot
```

Dies ist eine gute Möglichkeit, zusätzlich zur Objektbeschriftung (siehe *Beschriftungen*) zu überprüfen, ob die astrometrische Lösung und das Bild übereinstimmen.

Technische Informationen

Für die Photometrie verwendet Siril nur den B-V-Index, oder für den Gaia-Katalog das effektive Temperaturfeld Teff), das Informationen über die Farbe des Sterns liefert. Die drei Bildkanäle werden dann skaliert, um die beste Farbdarstellung für alle Sterne im Bild zu erzielen.

Für weitere Informationen über den KStar-Binärdateityp siehe diese Seite und diese Diskussion über kstars-devel und einige Entwicklungshinweise in Siril hier und hier.

Sha1-Summen für die 4 Katalogdateien:

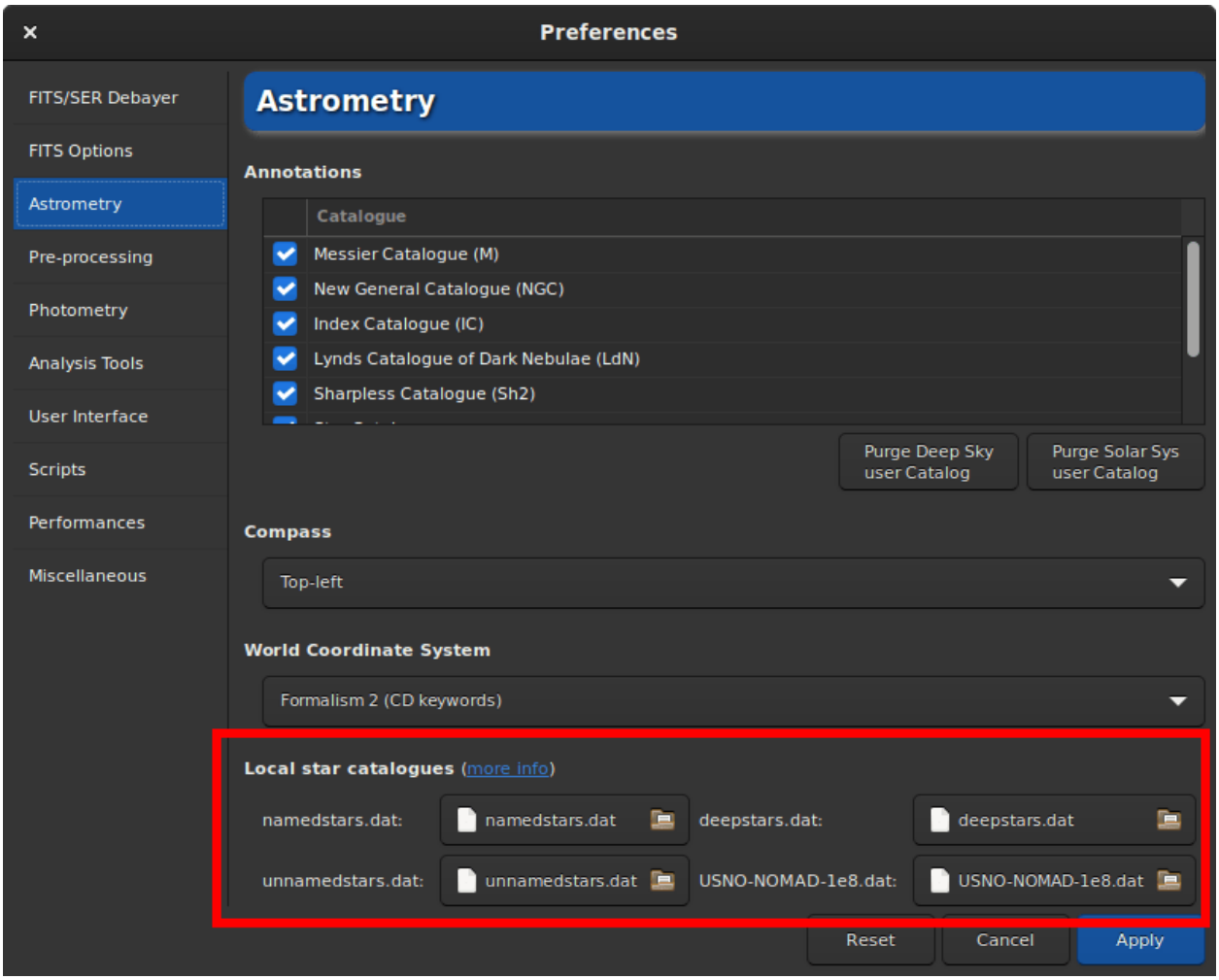


Abb. 4: Einstellungen bei lokalen Katalogen

```
4642698f4b7b5ea3bd3e9edc7c4df2e6ce9c9f7d  namedstars.dat
53a336a41f0f3949120e9662a465b60160c9d0f7  unnamedstars.dat
d32b78fd1a3f977fa853d829fc44ee0014c2ab53  deepstars.dat
12e663e04cae9e43fc4de62d6eb2c69905ea513f  USNO-NOMAD-1e8.dat
```

Lizenzen für die 4 Katalogdateien.

11.1.4 Verwendung des lokalen astronomy.net-Solvers

Installation

Seit Version 1.2 kann der Solver `solve-field` aus der astrometry.net Suite von Siril verwendet werden, um Bilder oder Bildsequenzen astrometrisch zu lösen.

Für Windows-Plattformen ist der einfachste Weg, es zu bekommen, ansvr zu verwenden. Wenn Sie das Standard-Installationsverzeichnis, d.h. *%LOCALAPPDATA%cygwin_ansvr*, nicht verändert haben, wird Siril ohne zusätzliche Einstellungen danach suchen. Wenn Sie cygwin haben und astrometry.net aus den Quellen gebaut haben, müssen Sie den Ort des cygwin root-Verzeichnisses in den *Eisntellungen* angeben.

Für MacOS folgen Sie bitte dieser Anleitung <http://www2.lowell.edu/users/massey/Macsoftware.html#Astrom>`_. Installieren Sie es mit homebrew und fügen Sie es dem `PATH` hinzu. Vergewissern Sie sich auch, dass das Programm für die Testbilder funktioniert, wie in der Anleitung angegeben, sowie außerhalb von Siril.

Bei Nicht-Windows-Betriebssystemen wird erwartet, dass die ausführbare Datei im `PATH` zu finden ist.

Die Verwendung dieses Tools ermöglicht es, Bilder *blind* zu lösen, ohne a priori zu wissen, welchen Bereich des Himmels sie enthalten. Es ist auch eine gute Alternative zu Sirils astrometrischem Plattenlöser, falls dieser versagt, da es sich um ein spezielles und bewährtes Werkzeug handelt, das auch die Feldverzerrung berücksichtigen kann.

Die Standardeinstellungen sollten in Ordnung sein, können aber mit dem Befehl *set* (Standardwerte in Klammern) oder in der Registerkarte *Astrometrie* in den Voreinstellungen geändert werden, wenn Sie das wirklich wollen. Wie groß der Bereich der erlaubten Skalierung ist (15%), wie groß der Radius der Suche von den Anfangskoordinaten ist (10 Grad), die Polynomordnung für die Feldverzerrung (0, deaktiviert), das Entfernen oder nicht der temporären Dateien (ja), die Verwendung des Ergebnisses als neue Standardbrennweite und Pixelgrößen (ja).

Indexdateien

Astrometry.net benötigt Indexdateien, um zu funktionieren. Wir empfehlen Ihnen dringend, die neuesten Indexdateien zu verwenden, die auf der Website <http://data.astrometry.net/>`_ verfügbar sind, d.h. die Serien 4100 und 5200. Das Bildfeld jeder Serie ist auf ihrer github page beschrieben. (die offizielle Dokumentation enthält diese Tabelle noch nicht).

Auf Unix-basierten Systemen können Sie einfach den Anweisungen in der Dokumentation folgen.

Unter Windows, wenn Sie ansvr verwenden, werden diese neuen Indexdateien vom Index-Downloader nicht zur Verfügung gestellt. Sie können sie dennoch separat herunterladen und dort speichern, wo die anderen Indexdateien aufbewahrt werden (ich würde empfehlen, die alten Dateien zu entfernen, auch wenn dies den Index-Downloader durcheinander bringen kann).

Wie es funktioniert

Genau wie der interne Plattenlöser wird Siril mit der Extraktion der Sterne aus Ihren Bildern fortfahren (um von der internen Parallelität zu profitieren) und diese Liste von Sternen an astrometry.net `solve-field` übermitteln. Wenn Sie dann möchten, dass astrometry.net den Index parallel durchforstet, müssen Sie dies in der Datei `astrometry.cfg` angeben.

11.1.5 Sequenzen lösen

Wenn eine Sequenz geladen wird, wird am unteren Rand des Dialogs ein zusätzliches GUI-Element angezeigt.

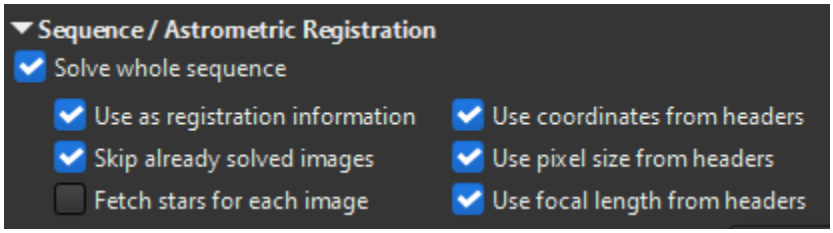


Abb. 5: Astrometrische Optionen bei Sequenzen

Sie können angeben, dass die gesamte Sequenz astrometrisch gelöst werden soll. Die bereits aufgelösten Bilder werden erneut gelöst, sofern das Kontrollkästchen *Bereits gelöste Bilder überspringen* nicht aktiviert ist. Um die *Astrometrische Registrierung* zu verwenden, müssen Sie die gesamte Sequenz verarbeiten, damit nützliche Informationen mit der Sequenz gespeichert werden (FWHM, Anzahl der Sterne, Hintergrundebene...). Lassen Sie daher das Kontrollkästchen *Als Registrierungsinformationen verwenden* aktiviert.

Wenn Sie Siril Solver mit lokalen Katalogen verwenden oder Astrometry.net verwenden, werden die im Dateiheader enthaltenen Informationen (falls vorhanden) verwendet, um das Bildzentrum und das Sampling für jedes Bild zu aktualisieren. Wenn Sie den Siril-Solver jedoch mit Online-Katalogen verwenden, wird standardmäßig ein einzelner Sternkatalog heruntergeladen, um zu viel Netzwerkverkehr und Serveranfragen zu vermeiden. Wenn die Bilder keine zu große Drift aufweisen und das Sampling gleich ist, ist dies normalerweise ausreichend. Wenn die Bilder jedoch nicht genügend Überlappung oder unterschiedliches Sampling aufweisen, können Sie das Herunterladen eines Sternkatalogs pro Bild erzwingen, indem Sie das Kontrollkästchen „Sterne für jedes Bild abrufen“ aktivieren.

Schließlich können Sie über die drei Felder auf der rechten Seite steuern, ob der Plattenlöser die Zielkoordinaten, die Pixelgröße und den Brennpunkt aus jedem Bildheader oder den im Dialogfeld angegebenen Werten lesen soll.

Am Ende der Sequenzlösung gibt das Protokoll an, wie viele Bilder gelöst wurden und ob welche übersprungen wurden.

```
Sequence processing succeeded.
Execution time: 676.35 ms
3 images successfully platesolved out of 3 included
(2 were already solved and skipped)
```

Das Lösen von Sequenzen ist auch über den Befehl *seqplatesolve* möglich.

> **Bemerkung**
>
> Beim Lösen von FITS-Sequenzen oder einer FITSEQ-Datei werden die Bilder direkt gespeichert, ohne dass eine neue Sequenz erstellt wird. Bei FITS-Sequenzen werden die Originaldateien nicht aktualisiert, wenn die Sequenz mithilfe symbolischer Links erstellt wurde. Stattdessen wird der Name des symbolischen Links verwendet, um eine neue FITS-Datei zu erstellen, wobei das Original unberührt bleibt. Beim Lösen einer SER-Sequenz wird eine neue Sequenz mit dem Präfix „**ps_**“ erstellt, da SER keine WCS-Daten speichern kann.

11.1.6 Sternerkennung

Standardmäßig verwendet die Sternsuche den Algorithmus *findstar* mit den aktuellen Einstellungen. Er funktioniert sehr gut, um viele Sterne zu finden, aber in manchen Fällen möchten wir die Sterne manuell erkennen oder einfach sehen, welche verwendet werden. Ein erster Schritt wäre, das Fenster *PSF* zu öffnen und die Sternsuche zu starten, dann die Einstellungen anzupassen (siehe die zugehörige Dokumentation *Dokumentation*).

Eine andere Möglichkeit wäre, die Sterne einzeln auszuwählen, indem man sie mit einem Auswahlrechteck umgibt und dann mit der rechten Maustaste auf *Stern auswählen* klickt. Je mehr Sterne ausgewählt werden, desto größer ist die Wahrscheinlichkeit, dass der Algorithmus erfolgreich ist.

Öffnen Sie dann im Astrometrie-Fenster den Abschnitt "Sternsuche" und aktivieren Sie die Option *Manuelle Suche*. Anstatt *Sternsuche* auszuführen, wird nun die aktuelle Liste der Sterne verwendet.

Bemerkung

Bei der Erkennung wird für RGB-Bilder immer die grüne Ebene verwendet. Bei CFA-Bildern ohne De-Bayering wird die grüne Ebene direkt extrahiert und stattdessen verwendet.

11.1.7 Verstehen der Ergebnisse

Wenn eine astrometrische Lösung gefunden wird, können wir auf der Registerkarte Konsole diese Art von Meldungen sehen:

```
Up is -5.26 deg CounterClockWise wrt. N
Resolution:      3.051 arcsec/px
Focal length:  254.21 mm
Pixel size:      3.76 µm
Field of view:    04d 51m 58.27s x 03d 01m 1.21s
Image center: alpha: 21h02m02s, delta: +68°10'48"
Was 119.64 arcmin from initial value
Saved focal length 254.21 and pixel size 3.76 as default values
Flipping image and updating astrometry data.
```

Die astrometrische Lösung liefert uns die J2000-Äquatorialkoordinaten des Bildzentrums, die projizierte horizontale und vertikale Dimension des Bildes am Himmel, die Brennweite, die dieses Feld für die gegebene Pixelgröße ergeben könnte, und folglich die tatsächliche Samplingrate, den Winkel des Bildzentrums macht mit der Erdachse, dem Sichtfeld und der Bildmitte. Es gibt auch an, wie groß der Abstand zum ursprünglich angegebenen Mittelpunkt war.

Wenn die astrometrische Lösung fehlschlägt, überprüfen Sie, ob die Startkoordinaten und die Pixelgröße korrekt sind, und versuchen Sie, die Größe zu ändern. Dadurch ändert sich die Anzahl der aus den Katalogen heruntergeladenen Sterne, und möglicherweise werden mehr Sterne identifiziert. Wenn Sirils Löser keine Lösung findet, ist es dennoch möglich, ein externes Tool zu verwenden. Die Lösung wird in jedem Fall in den *FITS-Header* geschrieben.

11.1.8 Visualisierung von Verzerrungen

Um die Gültigkeit der Lösung zu überprüfen, können Sie den Befehl *conesearch* verwenden. Dieser Befehl zeigt die Positionen der Sterne aus den Katalogen an und überprüft, ob sie mit den tatsächlichen Sternen im Bild übereinstimmen.

Die beiden folgenden Bilder zeigen Beschriftungen in der oberen rechten Ecke eines Bildes mit erheblicher Verzerrung. Das obere Bild ist die lineare Lösung, während das untere Bild unter Berücksichtigung von kubischen Polynomen gelöst wurde.

Wenn das Lösen mit Verzerrungen erfolgreich ist, kann mit dem Befehl *disto* oder dem Menü *Tools* → *Bildanalyse* → *Verzerrungen anzeigen* eine Darstellung der Korrekturen als Overlay auf dem Bild angezeigt werden.

Siril Kommandozeile

```
disto [clear]
```

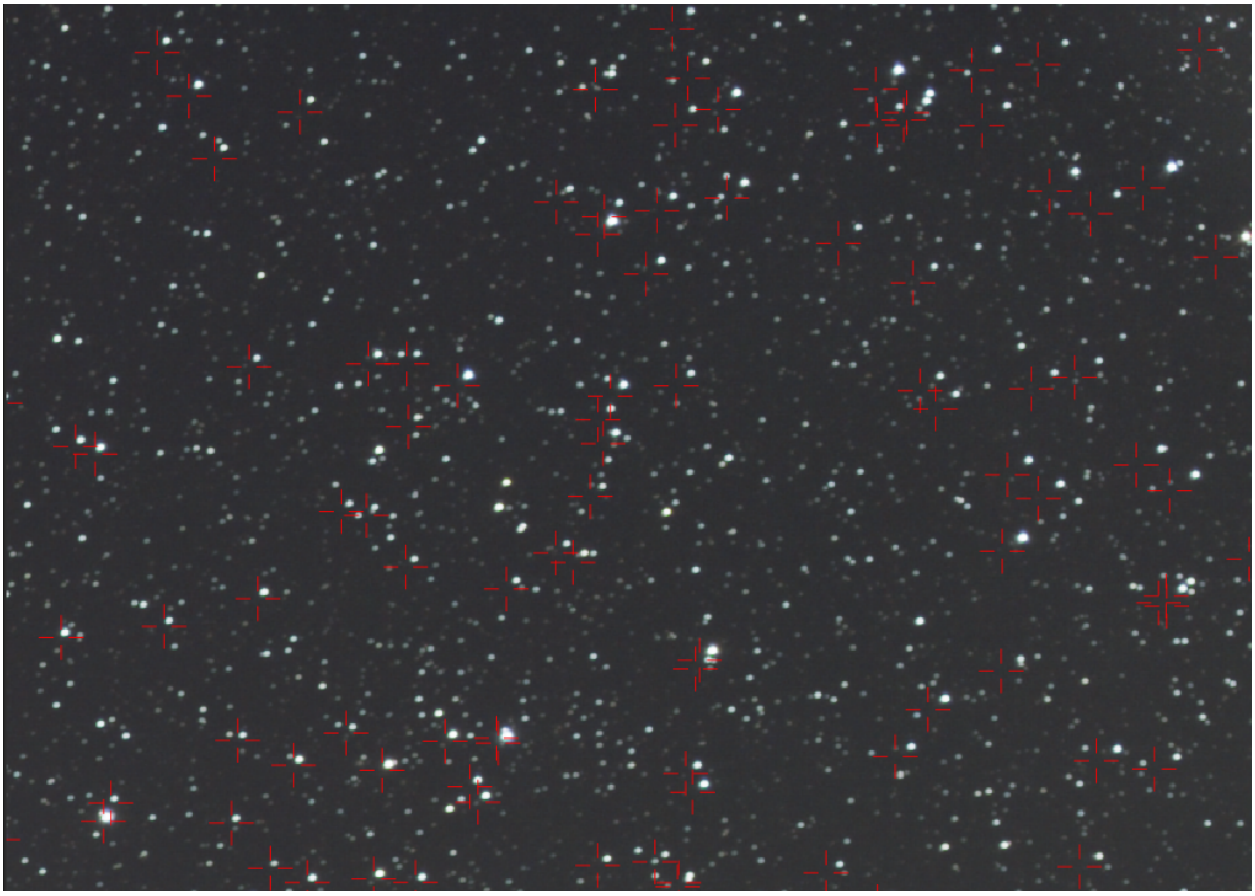


Abb. 6: Sternbeschriftungen mit linearer Lösung

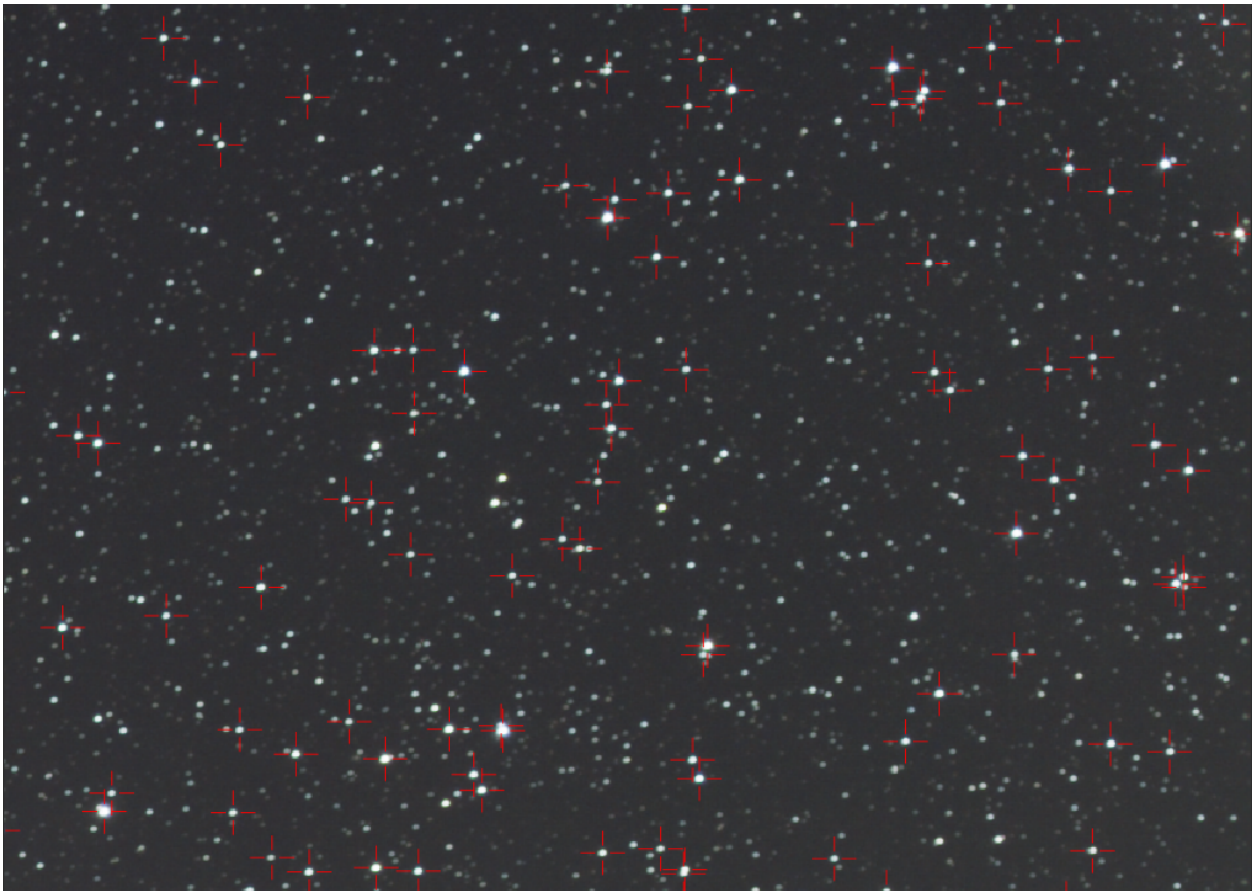


Abb. 7: Sternbeschriftung mit kubischer Lösung

Zeigt das Verzerrungsfeld eines astrometrisch gelösten Bildes mit Verzerrungstermen an

Option **clear** angeben zum deaktivieren

Ein Beispiel wird unten gezeigt.

Abb. 8: Bild mit Verzerrungs-Overlay

Sirilpy Skript

If you want to visualize the distortion field in 3D, you can use the Python script `Distortion3D.py` in the *Scripts->Python Scripts* menu. You will need to enable getting scripts from the *scripts repository*.

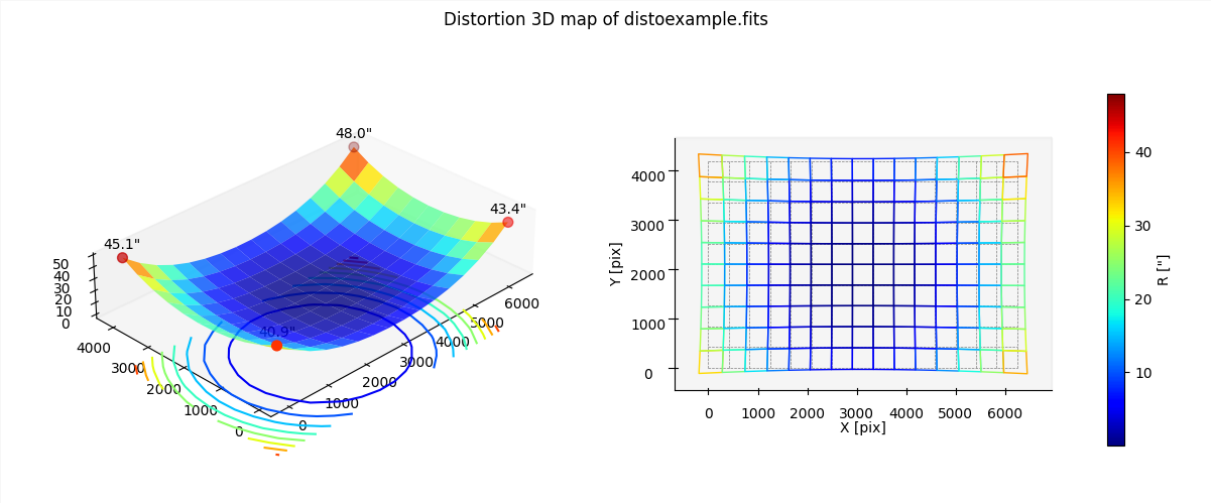


Abb. 9: 3D Verzerrungskarte

11.2 Beschriftungen/Objektnamen

Beschriftungen/Objektnamen sind Glyphen, die über den Bildern angezeigt werden, um das Vorhandensein bekannter Himmelsobjekte wie Galaxien, helle Sterne usw. darzustellen. Sie stammen aus Katalogen, können aber nur auf Bildern angezeigt werden, von denen wir wissen, welchen Teil des Himmels sie repräsentieren, also auf Bildern, für die eine **astrometrische Lösung** ermittelt wurde (Plate Solved) und die Weltkoordinatensystem-Informationen (WCS) in ihrem Header enthalten, also nur *FITS* oder *Astro-TIFF* Dateien.

Die astrometrische Lösung (Plate Solving) kann in Siril über den Eintrag *Werkzeuge→ Astrometrie → Astrometrische Lösung...* oder mit externen Tools wie astrometry.net oder ASTAP erfolgen.

Wenn ein astrometrisch gelöstes Bild in Siril geladen wird, sehen sie die Himmelskoordinaten für das Pixel unter dem Mauszeiger in der rechten unteren Ecke, und die Schaltflächen für die Anzeige von Beschriftungen sind verfügbar. Die erste Schaltfläche schaltet die Objektbeschriftungen um, die zweite das Himmels-Koordinatensystem und den Kompass.

11.2.1 Offline-Beschriftungskataloge

Siril wird mit einer vordefinierten Liste von Katalogen für die Beschriftungen geliefert:

- Messier Katalog (M)
- New General Catalogue (NGC)
- Index Katalog (IC)
- Lynds Katalo of Dark Nebulae (LdN) - Lynds Katalog der Dunkelnebel
- Sharpless Katalog (Sh2)
- Sternkatalog (3661 der hellsten Sterne)
- IAU Sternbildlinien
- IAU constellations names (positions collected from this page)

Zusätzlich können 2 *benutzerdefinierte Kataloge* verwendet werden:

- Benutzerkatalog DeepSky (DSO)
- Benutzerkatalog "Objekte des Sonnensystems" (SSO)

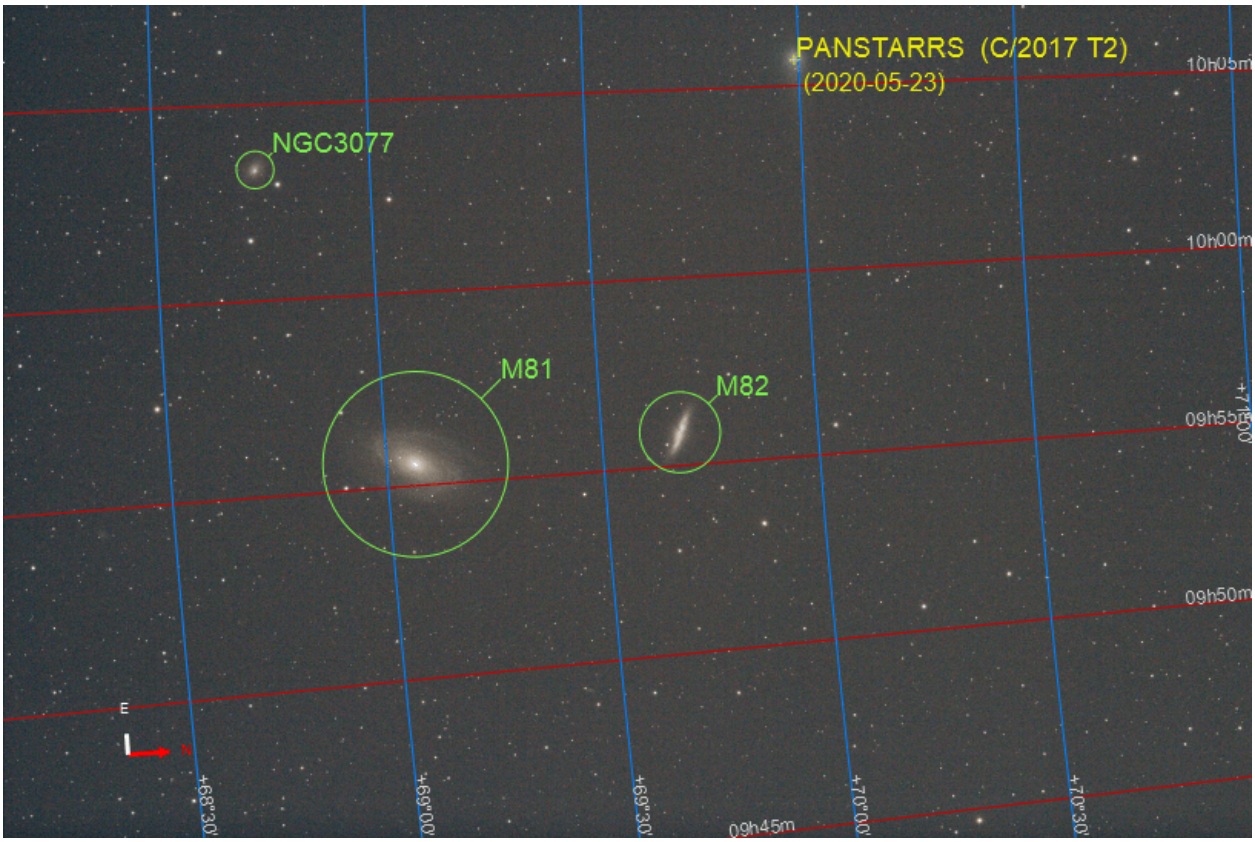


Abb. 10: Ansicht eines vollständig beschrifteten Bildes

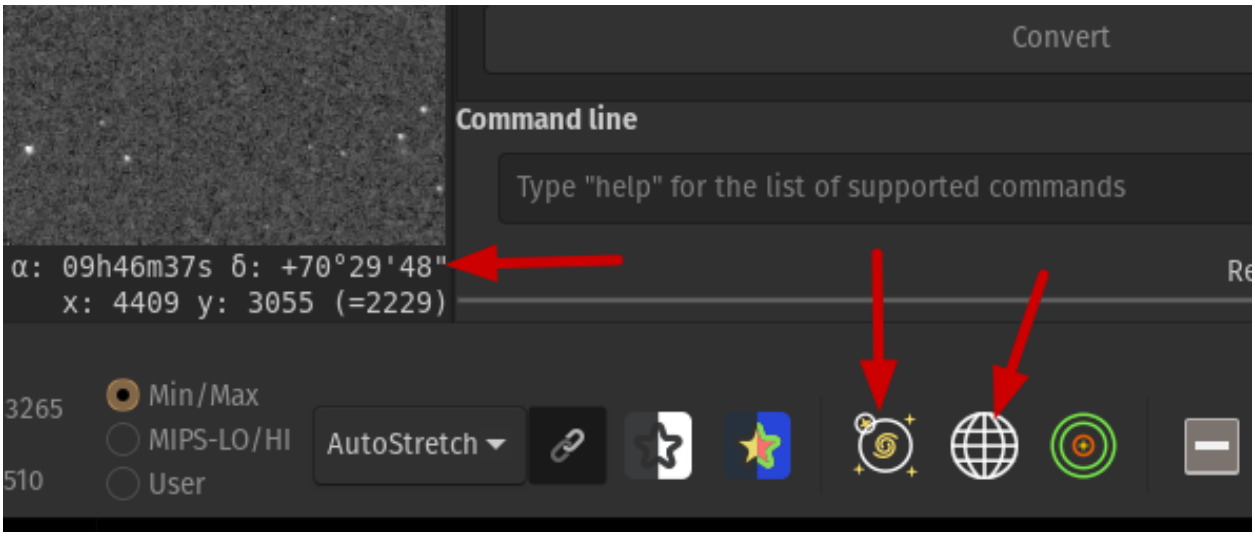


Abb. 11: Schaltflächen für Beschriftungen

Sie werden mit den Befehlen gefüllt, die im Abschnitt über *searching for a known object* beschrieben sind.

Sie werden im Verzeichnis der Benutzereinstellungen gespeichert. Ihr Speicherort hängt vom Betriebssystem ab:

- für *Unix-basierte* Betriebssysteme werden sie in `~/.config/siril/catalogue` sein
- unter *Windows* befinden sie sich in `%LOCALAPPDATA%\siril\catalogue`.

Alle obigen Kataloge können in den Einstellungen *Einstellungen* → *Astrometrie* im *Reiter* ein- und ausgeschaltet werden.

Die beiden *benutzerdefinierten Kataloge* können auch über die entsprechenden Schaltflächen geleert(d. h. gelöscht) werden.

Eine Bildlaufleiste auf der rechten Seite erlaubt es einfach durch die Katalogliste zu scrollen.

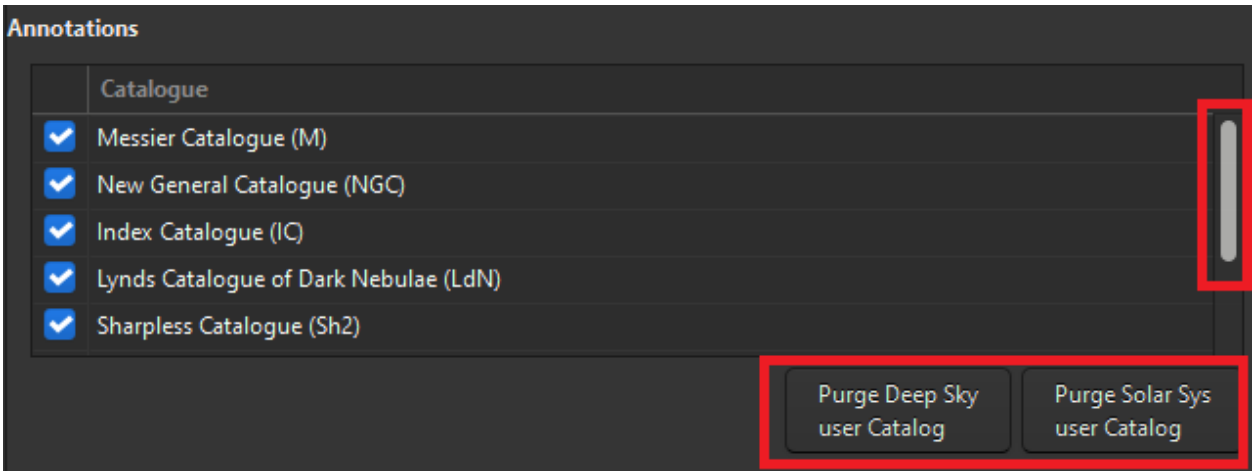


Abb. 12: Verwaltung der Kataloge in Einstellungen/Astrometrie

There's also a simpler way of accessing this list, as well as the button for displaying/hiding catalog annotations. Right-click on the annotations button in the toolbar and the list will be displayed in a popup window.

Diese Beschriftungskataloge werden in erster Linie zu Anzeigezwecken verwendet. Ab Siril 1.3 werden sie auch verwendet, um das Zentrum des Bildes für Astrometrie- oder Photometrie-Tools zu lokalisieren. Wenn das Objekt lokal gefunden wird, wird der Plattenlöser als lokal angezeigt. Wenn nicht, wird auf einen Online-Plattenlöser zurückgegriffen.

11.2.2 Online-Beschriftungskatalog

Möglicherweise möchten Sie andere Datenbanken als die bereits mit Siril gelieferten abfragen, die im Abschnitt *Offline Beschriftungskataloge* beschrieben sind. Dies funktioniert wieder nur für astrometrisch gelöste Bilder Bilder.

Ab Siril 1.3 ist dies mit dem Befehl *conesearch* möglich. Dieser neue Befehl ersetzt und erweitert die Funktionen, die zuvor von `nomad` und `solsys` ab Siril 1.2 bereitgestellt wurden.

Siril Kommandozeile

```
conesearch [limit_magnitude] [-cat=] [-phot] [-obscode=] [-tag={on|off}] [-log=
↪{on|off}] [-trix=] [-out=]
```

Zeigt standardmäßig Sterne aus dem lokalen Katalog für das geladene, astrometrisch gelöste Bild an, bis zur

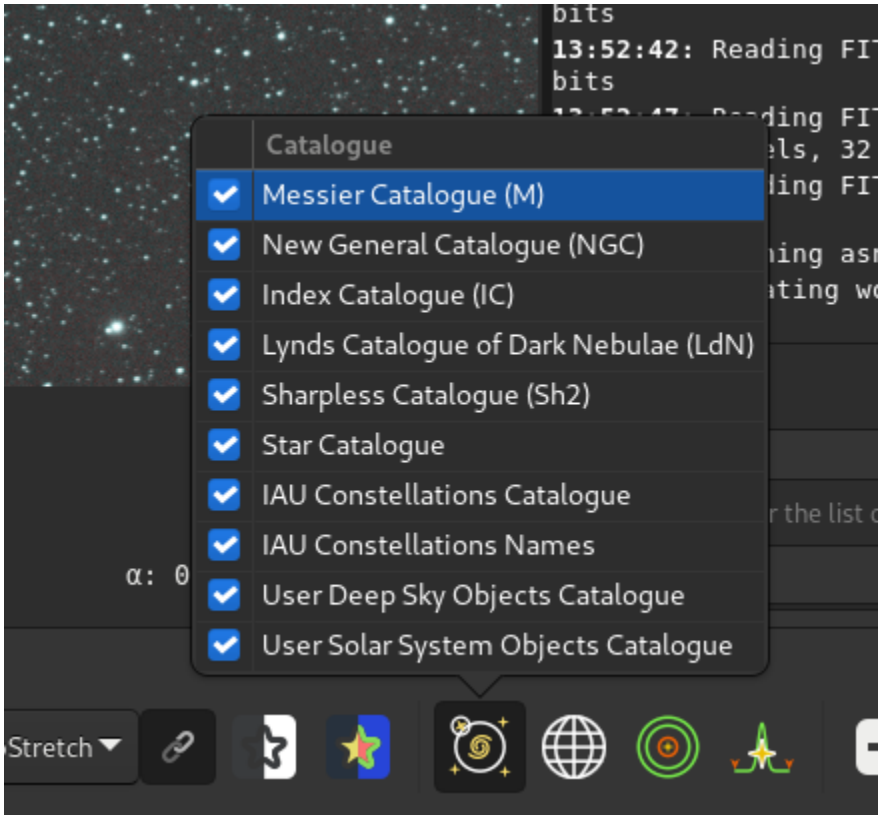


Abb. 13: Catalogue management in the pop-up window triggered by right-clicking on the annotation button.

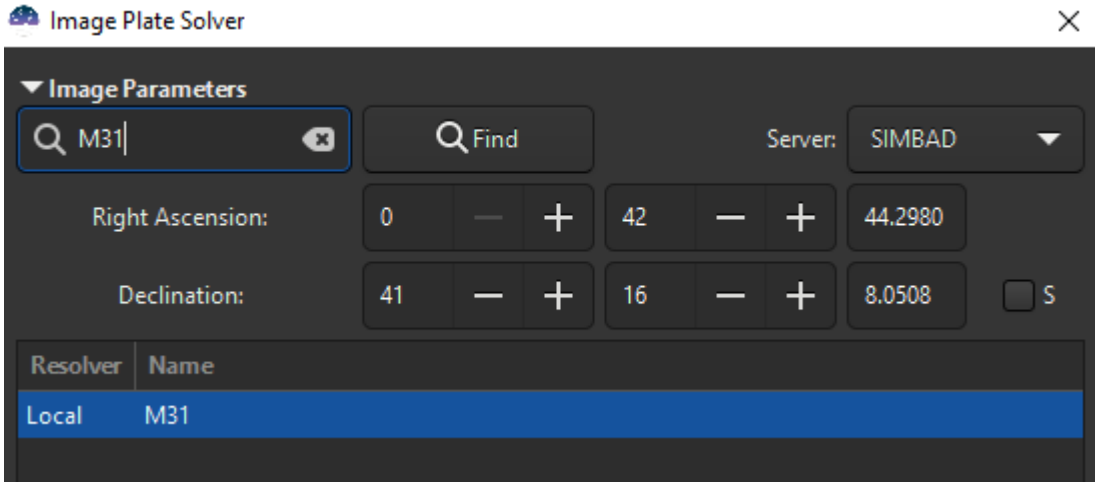


Abb. 14: Objekt im lokalen Beschriftungskatalog gefunden

angegebenen Grenzgröße (**limit_magnitude**) (standardmäßig 13 für die meisten Kataloge, außer 14.5 für aavso_chart, 20 für solsys und ohne Grenzgröße für pgc).
Ein alternative Onlinekatalog kann durch Angabe von **-cat=** angegeben werden mit folgenden Werten
- for stars: tycho2, nomad, gaia, localgaia, ppmxl, bsc, apass, gcvs, vsx, simbad, aavso_chart
- for exoplanets: exo
- for deep-sky: pgc
-für Objekte des Sonnensystems: solsys (der nächstgelegene`IAU observatory code <https://vo.imcce.fr/webservices/data/displayIAUObsCodes.php>`__ kann mit dem Argument **-obscode=** für eine bessere Positionsgenauigkeit übergeben werden)

Für Sternkataloge die photometrische Daten enthalten, werden Sterhen ohne B-V-Informationen beibehalten; sie können durch Angabe von **-phot** ausgeschlossen werden
Das Argument **-trix=** kann anstelle eines Katalogs gefolgt von einer Zahl zwischen 0 und 511 übergeben werden, um Sterne zu zeichnen, die im lokalen Katalog-Trixel der Ebene 3 enthalten sind (hauptsächlich für die Verwendung durch Entwickler)

Bei einigen Katalogen (bsc, gcvs, pgc, exo, aavso_chart und solsys) werden standardmäßig auch die Namen neben den Markern in der Anzeige (nur GUI) angezeigt und im Protokoll aufgeführt. Bei anderen Katalogen mit einer größeren Anzahl von Objekten, namentlich vsx und simbad, können diese Informationen ebenfalls angezeigt werden, aber da sie die Anzeige überladen könnten, sind sie standardmäßig nicht aktiviert. Dieses Verhalten kann mit den Optionen **-tag=on|off** zur Anzeige der Namen neben den Markern und **-log=on|off** zur Auflistung der Objekte im Konsolenprotokoll ein- und ausgeschaltet werden

Die Liste der im Bild vorhandenen Elemente kann optional durch Übergabe des Arguments **-out=** in einer CSV-Datei gespeichert werden

Auf diesen Befehl kann von der grafischen Benutzeroberfläche aus über *Werkzeuge→ Astrometrie → Beschriften…* zugegriffen werden.

In der nachstehenden Tabelle sind alle verfügbaren Kataloge sowie die Links zu den Originaldaten aufgeführt.

Tab. 1: Tabelle der verfügbaren Kataloge

Katalog	*-cat=*	Referenz
Lokal	*ausgelassen*	using *lokale Sternkataloge*
Tycho2	Tycho-2	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/297
Nomad	NOMAD	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/297
Gaia DR3	GAIA	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/355
PPMXL	PPMXL	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/317
Kataloge heller Sterne	BSC	https://cdsarc.cds.unistra.fr/viz-bin/cat/V/50
APASS DR9	APASS	https://cdsarc.cds.unistra.fr/viz-bin/cat/II/336
GCVS	GCVS	https://cdsarc.cds.unistra.fr/viz-bin/cat/B/gcvs
VSX	VSX	https://cdsarc.cds.unistra.fr/viz-bin/cat/B/vsx
SIMBAD	SIMBAD	https://simbad.cds.unistra.fr/simbad/sim-tap/
PGC/HYPERLEDA	PGC	https://cdsarc.cds.unistra.fr/viz-bin/cat/VII/237
Archiv für Expoplaneten	EXO	https://exoplanetarchive.ipac.caltech.edu
AAVSO VSP	AAVSO_Karte	https://app.aavso.org/vsp/
IMCCE Solar system skybot	SOLSYS	https://vo.imcce.fr/webservices/skybot/?conesearch

Die folgenden Abfragen werden Ihnen dank der folgenden Organisation zur Verfügung gestellt:

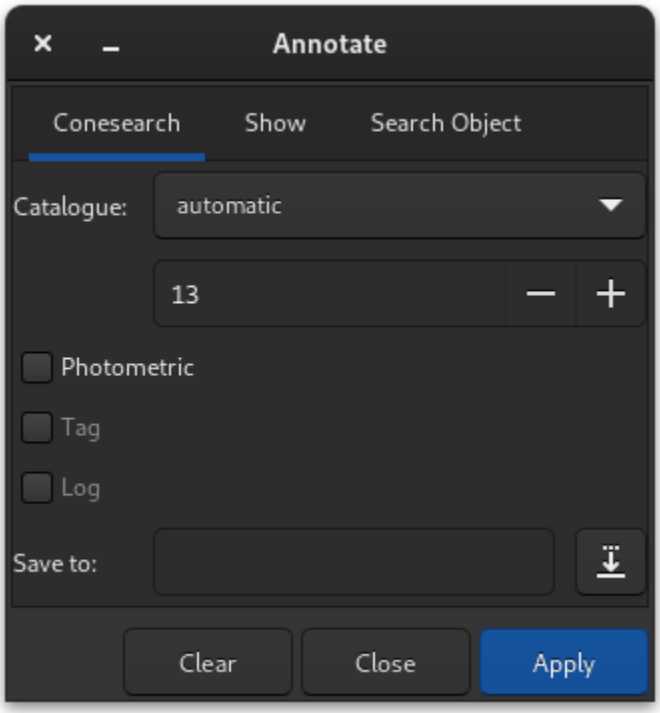


Abb. 15: Grafische Version des Befehls *conesearch*

- Vizier (über CDS Vizier TAP Abfrageservice)
- Simbad (über CDS Simbad TAP Abfrageservice)
- IMCCE (über seinen Skybot Kegel-Suchservice)
- NASA Exoplanetenarchiv (über seinen TAP Abfrageservice)

Bei Abfragen von Objekten im Sonnensystem können Sie einen zusätzlichen Parameter **-obscode=** übergeben, den 3-stelligen Code für ein IAU-Observatorium in der Nähe Ihres Beobachtungsstandorts. Dadurch wird die Genauigkeit der Beschriftungen verbessert. Bitte beachten Sie, dass die Ergebnisse dennoch leicht von denen abweichen können, die Sie erhalten, wenn Sie eine *direkte Ephemeridenabfrage für ein bestimmtes Objekt* durchführen, bei der der genaue Beobachtungsstandort verwendet wird (sofern im FITS-Header vorhanden).

Tipp

Ein bevorzugter Observatoriumscode kann im Reiter *Astrometrie* im Dialog *Einstellungen* festgelegt werden. Wenn festgelegt, wird dieser für das GUI-Tool *Objekte des Sonnensystems* und auch im Befehl *conesearch* verwendet, sofern nicht das Argument **-obscode=** angegeben wird.

Diese zusätzlichen Beschriftungen werden in *ROT* angezeigt, um sie von den Offline-Beschriftungen zu unterscheiden, die in *GRÜN* angezeigt werden. Diese Beschriftungen werden gelöscht, sobald die Schaltfläche `Objektnamen anzeigen` umgeschaltet wird.

11.2.3 Zusätzliche Kataloge

Möglicherweise möchten Sie Ihre eigenen Benutzerkataloge anzeigen. Dies können Sie mit dem Befehl *show* tun. Mit diesem Befehl können Sie beispielsweise auch CSV-Dateien anzeigen, die mit der Funktion zum Suchen von *Vergleichssternen* erstellt wurden.

Siril Kommandozeile

```
show [-clear] [{ -list=file.csv | [name] RA Dec }] [-nolog] [-notag]
```

Zeigt einen Punkt auf dem astrometrisch gelösten Bild unter Verwendung des temporären Katalogs der Benutzeranmerkungen, basierend auf seinen äquatorialen Koordinaten. Die Option **-clear** löscht diesen Katalog zuerst und kann allein verwendet werden.

Mehrere Punkte können über eine CSV-Datei mit der Option **-list=** übergeben werden, die mindestens Ra- und Dec-Spalten enthält. Wenn die übergebene Datei auch eine Spalte mit Namen enthält, werden die Namen als Tags im Bild verwendet und in der Konsole aufgeführt, es sei denn, sie werden mit den Optionen **-notag** und **-nolog** ausgeschaltet.

Diese Funktion ist nur über die GUI von Siril verfügbar

Diese Kataloge können jede *csv*-Datei (comma-separated) sein die folgende Regeln befolgen:

- Kommentarzeilen - falls vorhanden - sollten mit dem Zeichen # beginnen
- am Anfang sollte eine Zeile mit den Spaltennamen stehen, die durch Kommas getrennt sind
- Es sollten mindestens die Spalten *ra* und *dec* in Dezimalgraden angegeben werden.
- Die Spalten können in beliebiger Reihenfolge angegeben werden
- weitere Spalten können übergeben werden:
  - *Name* (text)
  - *Durchmesser* (Doppeltgenaue Fließkommazahl), der Objektdurchmesser in Bogenminuten
  - *mag* (doppeltgenaue Fließkommazahl), die Magnitude des Objekts
  - *Typ* (text), wird in der Konsole nach dem Namen in *()* angezeigt

Weitere Spalten können eingefügt werden, werden aber nicht genutzt.

Dieser Befehl ist über die grafische Benutzeroberfläche über *Werkzeuge→ Astrometrie → Beschriften...* zugänglich. Sie können entweder eine `*.csv`-Datei mit mehreren Zeilen kommagetrennter RA-, dec-Koordinaten in *CSV importieren* oder einen einzelnen Punkt in der Registerkarte *Einzelne Koordinaten* laden.

Bei einem Einzelpunkt können Sie das Ergebnis im *Deep Sky-Benutzerkatalog* speichern, indem Sie auf die Schaltfläche „Speichern“ klicken, wie in der folgenden Abbildung gezeigt:

Tipp

Der Speichern-Button funktioniert nur, wenn das Feld *Anzeigename* ausgefüllt ist.

Tipp

Sie können einen Stern oder ein Objekt auswählen und seine Koordinaten mit der Schaltfläche *Koordinaten aus Auswahl abrufen* abrufen.

Liste der bekannten Benutzerkataloge:

Manchmal erstellen Benutzer ihre eigenen Kataloge, wir können versuchen, sie hier zu verlinken, um allen zu helfen.

- Variable Sterne, entnommen aus GCVS 5.1, besprochen hier auf Französisch, (`Dateilink`).

Warnung

Im Gegensatz zu den Anweisungen im verlinkten Thema ist es nicht empfehlenswert, den Benutzer-DSO-Katalog durch solche Dateien zu ersetzen. Von der Verwendung wird abgeraten, da einige von ihnen besonders groß sein können und jedes Neuzeichnen von Beschriftungen enorm verlangsamen würden.

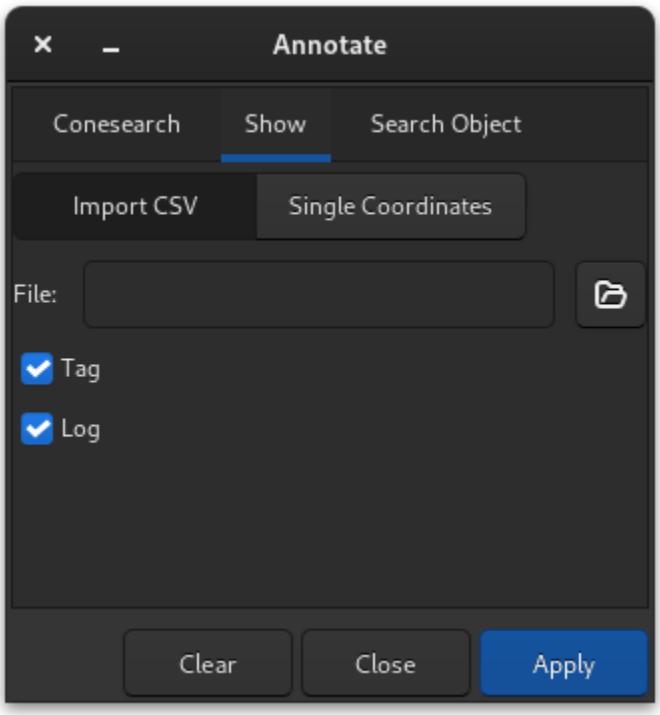


Abb. 16: Grafische Version des Befehls *show*

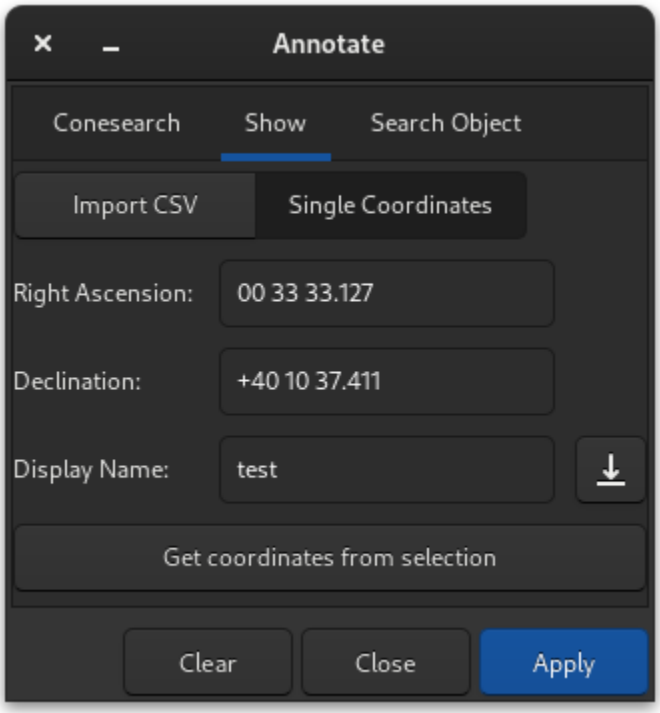


Abb. 17: Es besteht die Möglichkeit, nur einen einzelnen Punkt anzuzeigen und diesen im *Deep Sky Benutzerkatalog* zu speichern.

11.2.4 Suche nach einem bekannten Objekt

Wenn Sie wissen, dass sich ein bestimmtes Objekt irgendwo im Bild befindet (falls nicht, siehe den Abschnitt *Suche nach einem unbekannten Objekt*), können Sie es zu den Beschriftungen hinzufügen.

Deep-Sky-Objekte

Laden Sie ein astrometrisch gelöstes Bild und geben Sie `Strg+Umschalt + /` oder *Werkzeuge --> Astrometrie --> Beschriftungen...* im Pop-up-Menü ein.

Es steht ein Suchdialogfeld (siehe Abbildung unten) zur Verfügung, in das Objektnamen eingegeben werden können.

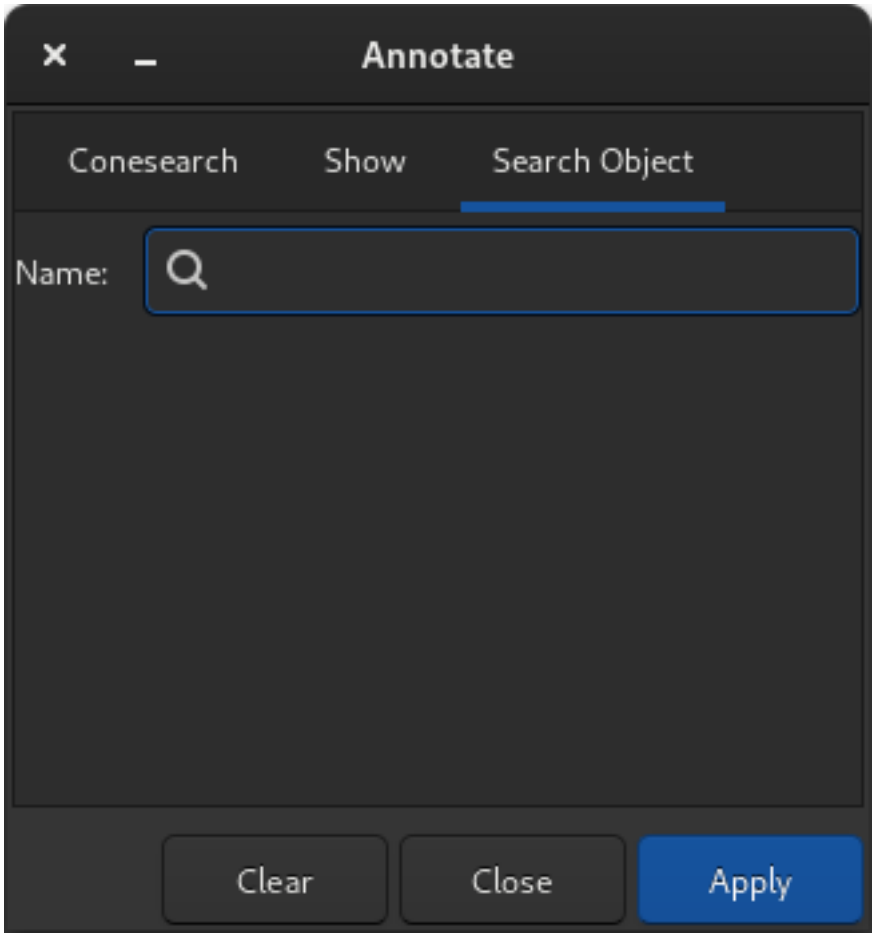


Abb. 18: Dialog Objekt suchen

Durch Drücken von `Enter` oder *Apply* wird zunächst in den vorhandenen Annotationskatalogen nach diesem Namen gesucht, falls er bereits unter einem anderen Namen vorhanden ist. Andernfalls wird eine Online-Anfrage an SIMBAD gesendet, um die Koordinaten eines Objekts mit diesem Namen abzurufen. Wenn das Objekt gefunden wird und noch nicht in einem Katalog vorhanden ist, wird es dem *Deep Sky-Benutzerkatalog* hinzugefügt.

Die Objekte dieses Katalogs werden in **ORANGE** angezeigt, während die Objekte aus den vordefinierten Katalogen in **GRÜN** dargestellt werden.

Beispiele für gültige Eingaben (Groß- und Kleinschreibung wird nicht berücksichtigt):

- `HD 86574` oder `HD86574` sind beide für diesen Stern gültig

Abb. 19: Deep-Sky-Objekte aus benutzerdefinierten und vordefinierten Katalogen

Objekte des Sonnensystems

Ab Siril Version 1.2 kann auch nach Objekten des Sonnensystems gesucht werden, indem der Dienst "Miriade ephemcc <https://ssp.imcce.fr/webservices/miriade/api/ephemcc/>" verwendet wird. Dies geschieht auf die gleiche Weise wie bei den Deep Sky Objekten, wobei dem Namen des zu suchenden Objekts ein Schlüsselwort vorangestellt wird, das die Art des Objekts angibt: `a:` für Asteroiden, `c:` für Kometen, `p:` für Planeten, `dp:` für Zwergplaneten und `` `s: `` für natürliche Satelliten. Wenn Sie ein Bild abfragen, das an einem Datum und zu einer Zeit aufgenommen wurde, die nahe genug an einem anderen Bild liegt (gleiche Nacht), das bereits mit SSOs versehen ist, werden deren zwischengespeicherte Positionen verwendet und um die Geschwindigkeit der einzelnen Objekte korrigiert, die von den Ephemeriden zurückgegeben wird. Die Objekte in diesem Katalog werden in **GELB** angezeigt.

Beispiele für gültige Eingaben (Groß- und Kleinschreibung wird nicht berücksichtigt):

- `c:67p` oder `c:C/2017 T2` sind gültige Formen für Kometen
- `a:1` und `a:ceres` sind beide gültig für *(1) Ceres*
- a:2000 BY4" ist gültig für *103516 2000 BY4*
- `p:4` oder `p:mars` können Beide für den Mars verwendet werden
- `dp:Pluto` ist gültig für Pluto
- `s:Moon` or `s:Io` ist gültig für natürliche Satelliten.

Warnung

Bilder, die keinen `DATE-OBS` Headerschlüssel haben, können nicht für SSOs beschriftet werden. Bilder, die keine Informationen über den Standort des Beobachters enthalten (SITELAT, SITELONG und SITEELEV Header Keys), werden trotzdem mit Beschriftungen versehen, allerdings unter der Annahme einer geozentrischen Beobachterposition, d.h. als ob man vom Erdmittelpunkt aus beobachten würde. Je nach der Entfernung der Objekte zur Erde kann dies dazu führen, dass die Positionen geringfügig von ihren tatsächlichen Positionen abweichen.

Befehl *catsearch*

Die gleiche Funktion ist über den Befehl *catsearch*:

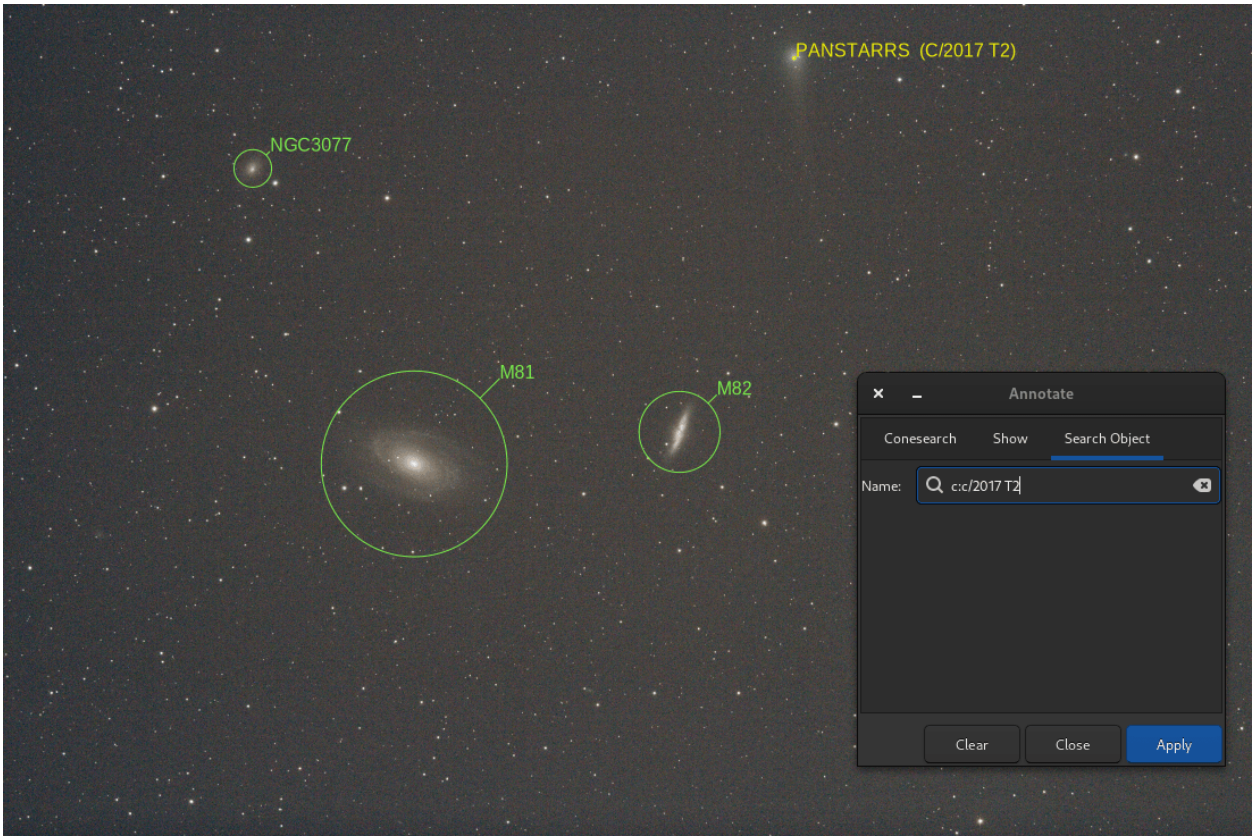


Abb. 20: Ergebnis eines Prozesses zur Sonnensystem-Suche

Siril Kommandozeile

```
catsearch name
```

Sucht ein Objekt nach **Name** und fügt es dem Benutzerkatalog für Anmerkungen hinzu. Das Objekt wird zunächst in den Beschriftungskatalogen gesucht, wenn es nicht gefunden wird, wird eine Anfrage an SIMBAD gestellt.

Das Objekt kann ein Objekt des Sonnensystems sein. In diesem Fall ist vor dem Objektnamen ein Präfix "a:" für Asteroid, "p:" für Planet, "c:" für Komet, "dp:" für Zwergplanet oder "s:" für natürlicher Satellit erforderlich. Die Suche erfolgt nach dem Datum, der Uhrzeit und dem Beobachtungsort, die in der Kopfzeile des Bildes angegeben sind, unter Verwendung des "IMCCE Miriade Service <https://ssp.imcce.fr/webservices/miriade/howto/ephemcc/#howto-sso>"

11.2.5 Suche nach einem unbekannten Objekt

Besonders nützlich für photometrische Arbeiten ist die Möglichkeit, einen Stern oder andere Objekte im Bild zu identifizieren, indem man eine Auswahl um sie herum zieht, mit der rechten Maustaste klickt, um das Kontextmenü aufzurufen, und den Eintrag *PSF* auswählt.

Dadurch wird das PSF-Fenster geöffnet, und wenn es sich um einen Stern handelt, werden die Gauß'schen Anpassungsparameter angezeigt, aber es wird auch ein Weblink unten links im Fenster angezeigt. Wenn Sie diesen Link öffnen, gelangen Sie auf die SIMBAD-Seite <https://simbad.cds.unistra.fr/simbad/sim-fcoo>`_ mit den Koordinaten des Objekts und in vielen Fällen auch mit dem Namen des Objekts.

SIMBAD enthält nicht alle bekannten Objekte, aber die Koordinaten auf der Seite können als Ausgangspunkt für die Suche nach dem Objekt in anderen Online-Katalogen verwendet werden, z. B. Gaia DR3 (VizieR).

KAPITEL 12

Dynamische Point Spread Funktion

Dieser Abschnitt beschreibt die beiden wesentlichen Schritte, die zur Erkennung von Sternen in Einzelbildern durchgeführt werden. Die Erkennung eines Einzelbildes kann mit *Werkzeuge → Bildinformationen → Dynamisches PSF* oder mit der Tastenkombination `Strg` + `F6` durchgeführt oder feinjustiert werden.

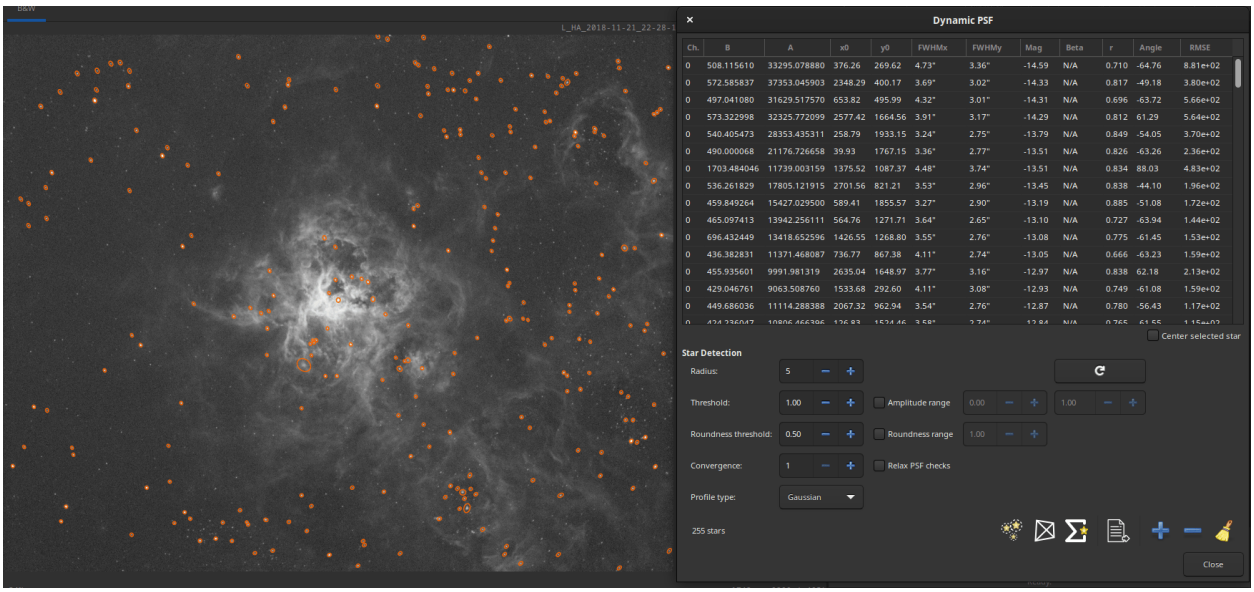


Abb. 1: Dynamische PSF auf einem Deep-Sky-Bild.

Das Verfahren ist wie folgt:

- eine erste Erkennung von potenziellen Sternkandidaten vornehmen
- Anpassung eines PSF-Modells für jeden Kandidaten. Anhand der Parameter der Modellanpassung eine Überprüfung durchführen, um sicherzustellen, dass es sich um einen Stern handelt, und Nicht-Stern-Kandidaten ablehnen.

Am Ende dieses Prozesses erhält man eine Liste von Sternen mit den Positionen im Bild relativ zur linken oberen Ecke und den gemessenen Helligkeit aller Sterne in der Liste.

12.1 Erste Sternkandidaten

Während es beim Betrachten eines Bildes offensichtlich erscheint, wo sich die Sterne befinden, ist es etwas schwieriger, den Prozess in mathematische Operationen und Kriterien zu übertragen. In diesem Abschnitt wird der zugrunde liegende Algorithmus kurz beschrieben. Er ist inspiriert von dem Handbuch der Software "DAOPHOT" DAOPHOT software manual [Stetson1987], wobei Vereinfachungen vorgenommen wurden, um die Leistung zu steigern. Der ursprüngliche Algorithmus zielte darauf ab, alle möglichen Sterne zu erkennen und diente der Erstellung von Sternkatalogen, während Siril Sterne hauptsächlich als Merkmale für die Registrierung erkennen muss. Außerdem muss der Algorithmus auf eine große Vielfalt unterschiedlicher Bilder reagieren können - die meisten von uns haben keine professionelle astronomische Ausrüstung im Hinterhof - und wir mussten einige Entscheidungen hinsichtlich der Einschätzung der Abbildungsbedingungen (Abtastung, Seeing usw.) treffen.

Im Laufe der Jahre hat sich unsere Implementierung zu dem entwickelt, was sie heute ist. Sie zielt darauf ab, keine sehr hellen Sterne zu übersehen, die für die Registrierung wichtig sind, und so viele Ausreißer wie möglich zurückzuweisen, während sie gleichzeitig vertretbar schnell bleibt.

Sie kann in folgende Schritte zerlegt werden:

- die Statistik des Bildes berechnen, um sowohl den Hintergrund als Median des Bildes als auch sein Rauschen zu erfassen. Dabei wird davon ausgegangen, dass das Bild eine relativ gleichmäßige Helligkeit hat. Folglich wird die Erkennung in den Ecken weniger effizient sein, wenn nach der Kalibrierung eine starke Vignettierung vorliegt.
- es wird auch der Dynamikbereich berechnet, definiert als das Maximum des Bildes abzüglich des Hintergrunds. Dies wird später nützlich sein, um gesättigte Sterne zu erkennen.
- das Bild mit einem Gauß-Kernel glätten. Die ideale Glättung wäre die Verwendung eines Kernels, der die gleiche FWHM wie das Bild hat. Stattdessen haben wir eine mittlere Größe gewählt, die in einem sehr breiten Spektrum von Bedingungen zufriedenstellende Ergebnisse liefert. Dies ermöglicht es, "blind" gegenüber den Bildparametern zu sein.
- auf der geglätteten Version des Bildes lokale Maxima über einem Pegel erkennen, der als Hintergrundpegel plus dem X-fachen des Rauschens definiert ist (X kann mit dem `threshold`-Wert in der GUI variiert werden). Stellen Sie sicher, dass dies ein Maximum über einer bestimmten Boxgröße ist (definiert durch den Parameter `radius`).
- es wird eine Plausibilitätsprüfung durchgeführt, um sicherzustellen, dass das Maximum und seine Nachbarn deutlich über den umgebenden Pixeln liegen (um z. B. Flecken in den hellen Teilen eines Nebels auszuschließen).
- Überprüfung, ob der Kern um die Maxima gesättigt ist, d.h. durchgängig nahe an der oberen Grenze des dynamischen Bereichs liegt. Wenn ja, wird ein Edge-Walking-Algorithmus ausgeführt, um die Grenze des gesättigten Bereichs zu ermitteln.
- Die erste und zweite Ableitung entlang einer horizontalen und vertikalen Linie , die durch den Mittelpunkt verläuft, werden verwendet, um den lokalen Hintergrund, die Amplitude und die Größe des Sterns in allen Richtungen (oben, unten, links und rechts) zu ermitteln.
- Wenn die Parameter in allen Richtungen symmetrisch genug sind (bis hin zum Parameter `roundness`), wird der Stern als potentiellen Kandidat bestätigt.

Sobald die Liste der potenziellen Kandidaten zusammengestellt ist, werden sie nach abnehmender Amplitude sortiert und dem PSF-Anpassungsalgorithmus zugeführt, der im Abschnitt *Minimalisierung* beschrieben ist.

12.2 Modelle

Im dynamische PSF-Fenster werden zwei Modelle verwendet. Im Allgemeinen ist das Moffat-Modell viel besser geeignet, um Objekte wie Sterne anzupassen.

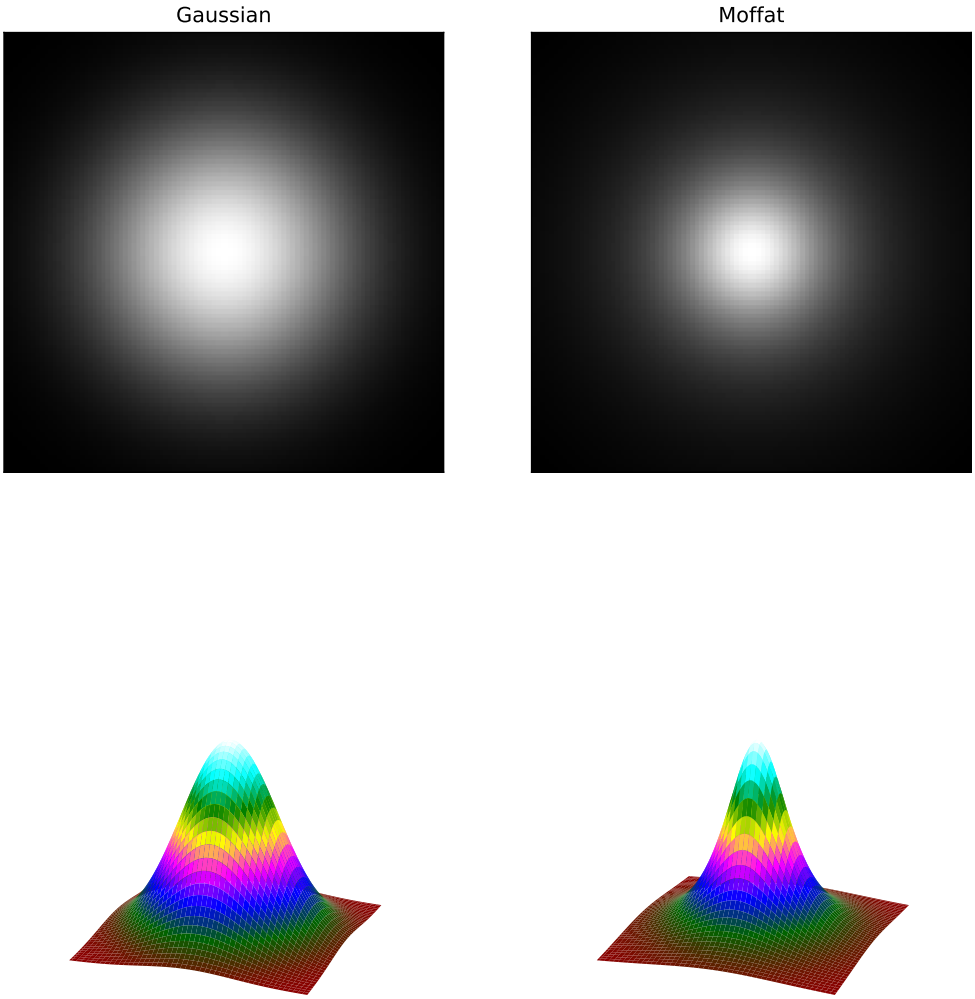


Abb. 2: Darstellung zweier kreisförmiger PSFs nach einem Gauß-Profil und einem Moffat-Profil. Beide Modelle verwenden die gleichen Parameter, wobei das Moffat-Profil $\beta = 1.4$ verwendet.

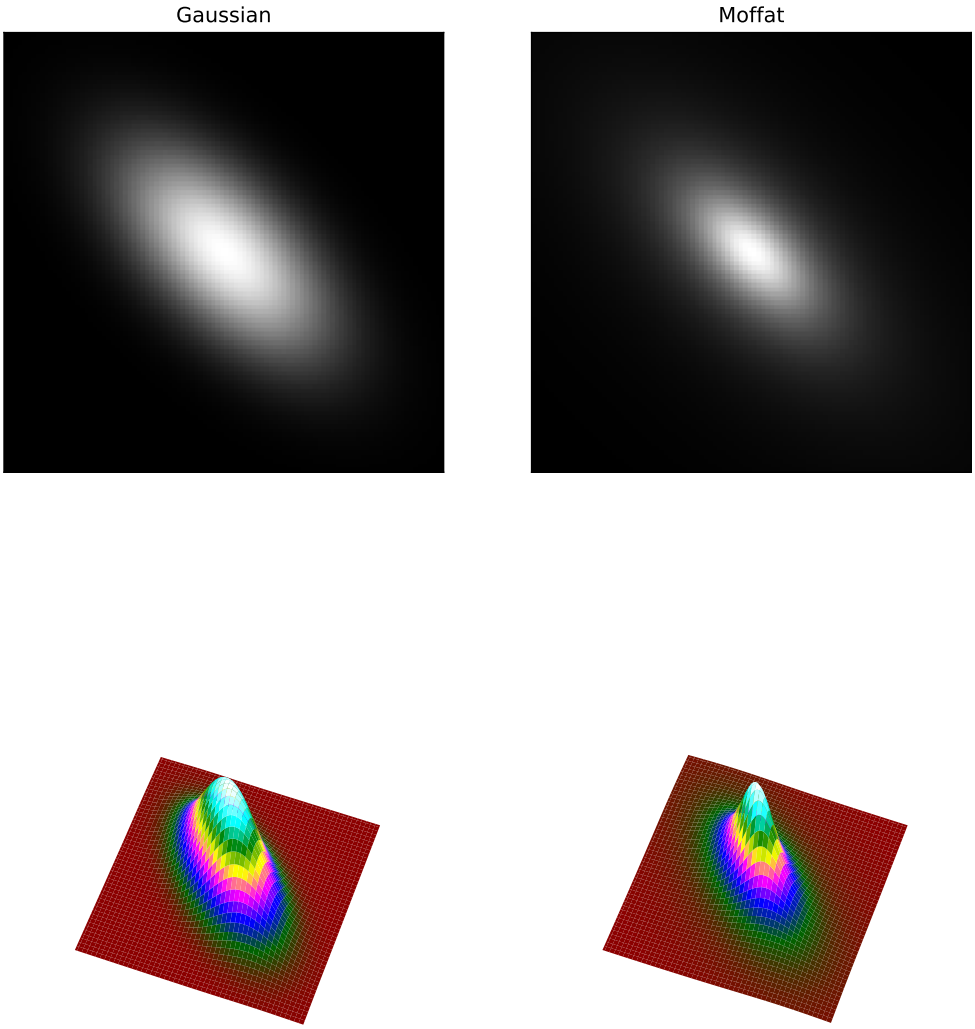


Abb. 3: Die gedrehte Gauß- und die Moffat-Funktion haben $\sigma_x = 2\sigma_y$, $\theta = 45$. Für Moffat gilt: $\beta = 1.4$.

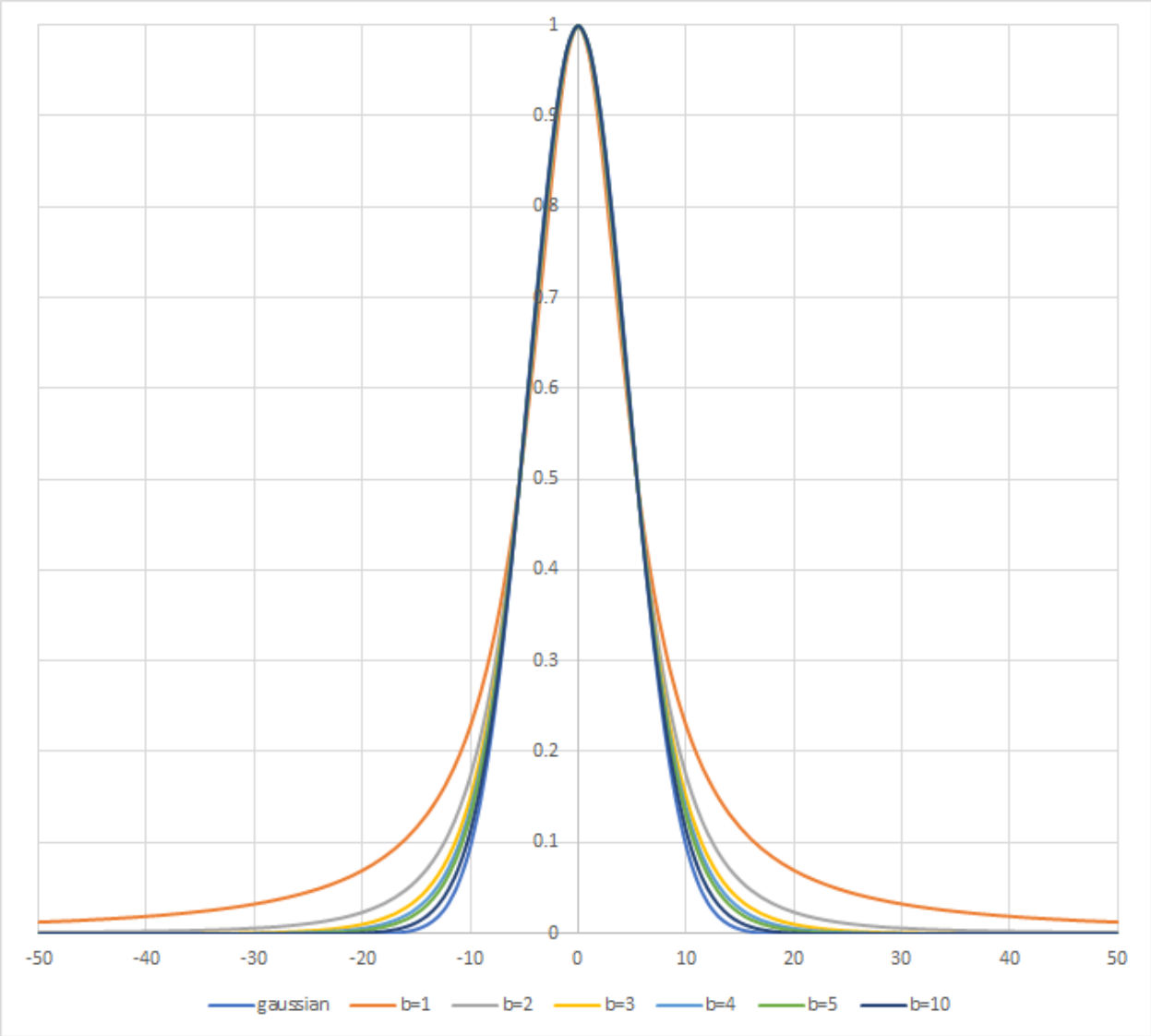


Abb. 4: Sternprofil mit Gauß- und Moffat-Modell. Es werden mehrere β-Werte ausprobiert. $\beta = 10$ ergibt ein Profil, das dem Gaußschen Profil sehr nahe kommt.

Theorie

1. Eine elliptische Gaußsche Anpassungsfunktion, definiert als

$$G(x,y) = B + Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)}. \tag{12.1}$$

2. Eine elliptische Moffat-PSF-Anpassungsfunktion, definiert als

$$M(x,y) = B + A\left(1+\frac{(x-x_0)^2}{\sigma_x^2}+\frac{(y-y_0)^2}{\sigma_y^2}\right)^{-\beta}, \tag{12.2}$$

wobei:

- B ist der durchschnittliche lokale Hintergrund.
- A ist die Amplitude, d.h. der Maximalwert der angepassten PSF.
- x_0, y_0 sind die Koordinaten des Schwerpunkts in Pixel-Einheiten.
- σ_x, σ_y sind die Standardabweichung der Gaußverteilung auf der horizontalen und vertikalen Achse, gemessen in Pixeln.
- β ist der Exponent aus der Moffat-Formel, der die Gesamtform der Anpassungsfunktion steuert. Die Obergrenze dieses Parameters wurde auf 10 festgelegt. Ein höherer Wert ist bedeutungslos und bedeutet, dass die Gaußfunktion gut genug ist, um den Stern abzubilden.

Andere Parameter werden von diesen angepassten Variablen abgeleitet:

- FWHM_x und FWHM_y: Das Full Width Half Maximum auf der X- und Y-Achse in Pixel-Einheiten. Diese Parameter werden wie folgt berechnet:
  - $\text{FWHM}_x = 2\sigma_x\sqrt{2\log 2}$.
  - $\text{FWHM}_y = 2\sigma_y\sqrt{2\log 2}$.
  - Es ist möglich, die FWHM-Parameter in der Einheit Bogensekunden zu erhalten. Dazu müssen Sie alle Felder, die Ihrer Kamera und Ihrem Objektiv/Teleskop-Fokus entsprechen, im Parameterfenster des Hamburger-Menüs ausfüllen, dann *Bildinformationen* und *Information...* ausführen. Wenn die Standard-FITS-Schlüsselwörter `FOCALLEN`, `XPIXSZ`, `YPIXSZ`, `XBINNING` und `YBINNING` aus dem FITS-Header gelesen werden, berechnet das PSF auch den Bildmaßstab in Bogensekunden pro Pixel.
- r: Der Rundheitsparameter. Er wird ausgedrückt als $\text{FWHM}_x/\text{FWHM}_y$, wobei $\text{FWHM}_x > \text{FWHM}_y$ die Symmetriebedingung ist.
- Ein weiterer Parameter wird ebenfalls sowohl im Gauß- als auch im Moffat-Modell angepasst. Dies ist der Drehwinkel $theta$, der im Bereich $[-90, +90]$ definiert ist. Die Hinzufügung dieses Parameters impliziert eine Koordinatenänderung, bei der die in (12.1) und (12.2) ausgedrückten Variablen x und y durch x und y ersetzt werden:

$$\begin{aligned} x' &= +x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta. \end{aligned} \tag{12.3}$$

12.3 Minimierung

Die Minimierung erfolgt mit einem nicht-linearen Levenberg-Marquardt-Algorithmus dank der sehr robusten GNU Scientific Library. Dieser Algorithmus wird verwendet, um das Minimum einer Funktion zu finden, die einen Satz von Parametern auf einen Satz von Beobachtungswerten abbildet. Es handelt sich um eine Kombination aus zwei

Optimierungstechniken: der Methode des Gradientenabstiegs und der invers-hessischen Methode.

Der Levenberg-Marquardt-Algorithmus passt den Kompromiss aus diesen beiden Methoden in Abhängigkeit von der Krümmung der zu minimierenden Funktion an. Wenn die Krümmung klein ist, verwendet der Algorithmus die Methode des Gradientenabstiegs, und wenn die Krümmung groß ist, verwendet der Algorithmus die inverse hessische Methode.

Seit Version 1.2.0 wird der gesättigte Teil des Sterns aus dem Anpassungsprozess entfernt, wodurch der nicht gesättigte Teil viel genauer erfasst werden kann. Dies ermöglicht die "Rekonstruktion" des Sternprofils bei Verwendung der Menüoption *Sterne entsätigen* oder des Befehls *unclipstars*.

12.4 Nutzung

Dynamische PSF kann auf zwei verschiedene Arten aufgerufen werden, je nachdem, was Sie wollen:

Vielleicht möchten Sie nur einen oder ein paar Sterne einpassen. In diesem Fall können Sie, nachdem Sie eine Auswahl um einen ungesättigten Stern gezogen haben (dies ist wichtig für die Genauigkeit des Ergebnisses), entweder mit der rechten Maustaste klicken und die Option *Stern auswählen* wählen, auf die Schaltfläche + im Dialogfeld "Dynamisches PSF" klicken oder `Strg` + `Leertaste` eingeben. Daraufhin wird eine Ellipse um den Stern gezeichnet. Um den Dialog zu öffnen, können Sie auch die Tastenkombination `Strg` + `F6` verwenden.

Sie können so viele Sterne wie möglich analysieren, indem Sie auf das Symbol klicken oder die Befehlszeile *findstar* verwenden. Alle gefundenen Sterne sind von einer Ellipse umgeben: orange, wenn der Stern in Ordnung ist, magenta, wenn der Stern gesättigt ist. Es ist auch möglich, den Durchschnitt der berechneten Parameter anzuzeigen, wie unten dargestellt, indem man auf die Schaltfläche klickt.

Die Erkennung von Sternen hat eine Reihe von Anwendungen:

- Siril verwendet es intern für astrometrische Zwecke bei der Registrierung von Bildsequenzen. Dies geschieht automatisch und erfordert keinen Benutzereingriff.
- Da Sterne im Vergleich zu lichtschwachen Objekten wie Nebeln oder Galaxien sehr hell sind, kommt es häufig vor, dass einige Sterne in einem Bild gesättigt sind, was bedeutet, dass ihr Helligkeitsprofil beschnitten ist. Dies kann bei einigen Bildverarbeitungsfunktionen, insbesondere bei der Dekonvolution, Probleme verursachen und führt zu einem Verlust von Farbinformationen und einer etwas größeren Aufblähung der Sterne bei der Anwendung von Streckungen. Die Analyse aller Sterne zeigt Ihnen, welche Sterne gesättigt sind, und Sie können dann die Menüoption *Sterne entsättigen* oder den Befehl *unclipstars* verwenden, um das Problem durch Synthese des abgeschnittenen Teils des Profils zu beheben.

  Siril Kommandozeile

  ```
  unclipstars
  ```

  Reprofiliert übersättigte Sterne, um sie zu entsättigen, und skaliert die Ausgabe so, dass alle Pixelwerte <= 1,0 sind

- Im Idealfall sollten alle Sterne in einem Bild vollkommen rund sein. Probleme wie Koma, Astigmatismus und schlechte Nachführung sowie Probleme wie ein falscher Backfokus bei Korrektoren können jedoch zu verzogenen, elliptischer Sterne in einem Bild führen. Die Ellipsen, die mit dem Werkzeug Dynamische PSF erzeugt werden, können solche Probleme visuell gut veranschaulichen.
- Die Untersuchung der durchschnittlichen Sternparameter, insbesondere FWHM und der Beta-Parameter der Moffat-Anpassungsfunktion, kann Informationen über die Qualität des Seeings in einem Bild liefern.

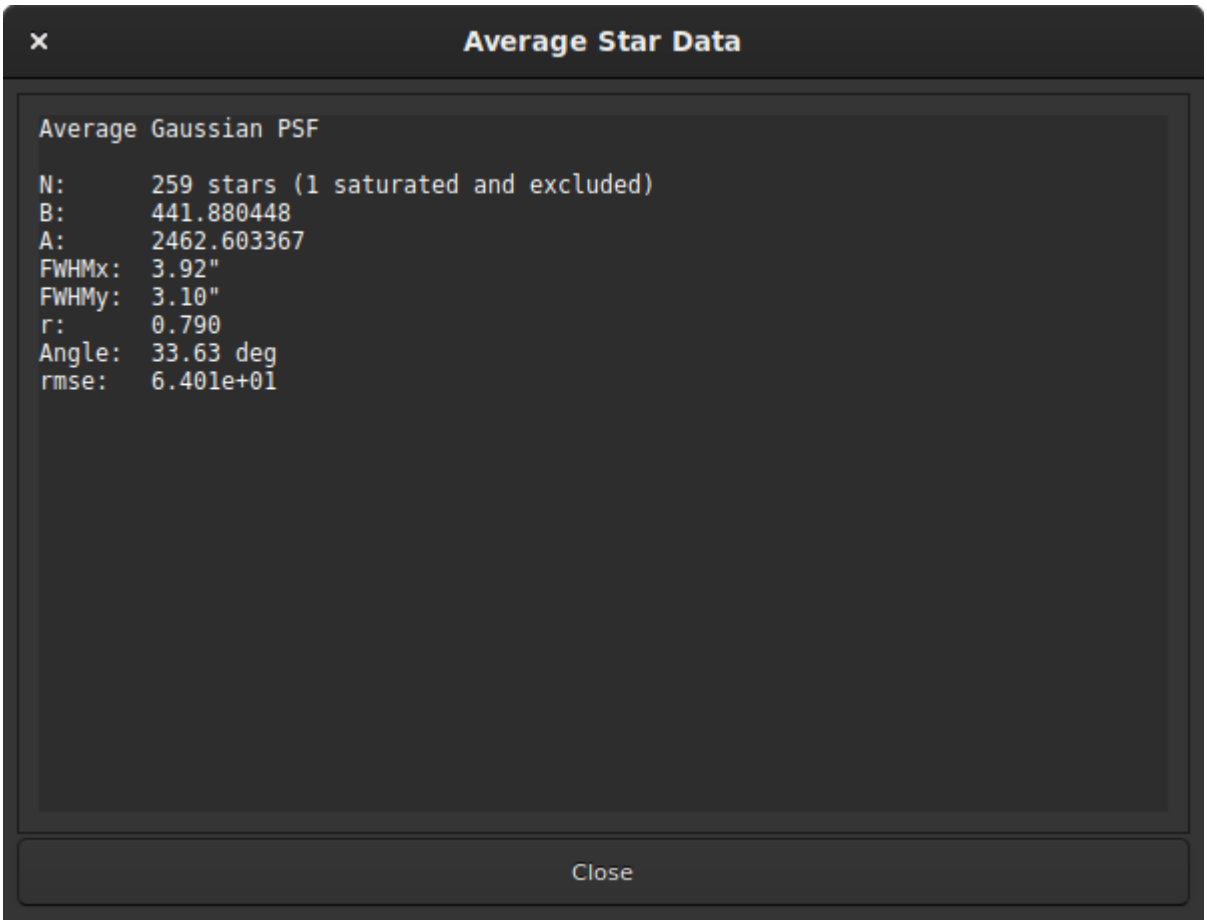


Abb. 5: Durchschnitt der angepassten Sterne im Gaußschen Modell.

- Die Erkennung aller Sterne ist der erste Schritt zur Verwendung des Werkzeugs *Sternbearbeitung → Stern-Neuzusammensetzung*. Dieses Werkzeug erstellt korrigierte Helligkeitsprofile für alle erkannten Sterne und kann zur Erstellung einer synthetischen Sternmaske verwendet werden, die dann mit einem von Starnet++ erzeugten sternlosen Bild gemischt werden kann, um andernfalls astrometrisch nicht lösbare Sterne in einem Bild zu korrigieren. In diesem Fall kann die Erkennung von Sternen unter Verwendung des Moffat-Profils ein realistischeres Ergebnis liefern und auch das Herausfiltern von Galaxien, die fälschlicherweise als Sterne erkannt wurden, erleichtern, indem die Einstellung Minimum beta verwendet wird.
- Die Schaltfläche **Ausgewählten Stern zentrieren** kann verwendet werden, um einen bestimmten Stern in der Liste schnell und einfach im Bild zu finden, indem er im Ansichtsfenster zentriert wird. Dies ist nützlich, wenn Sie alle Sterne erkannt haben und bestimmte Lösungen überprüfen möchten, um sicherzustellen, dass es sich wirklich um einen Stern und nicht um eine Galaxie oder einen kosmischen Strahl handelt.
- In ähnlicher Weise wird durch Klicken auf eine orange- oder magentafarbene Sternenellipse im Hauptfenster die ausgewählte Sternlösung im Dialogfeld Dynamische PSF hervorgehoben. Dies kann nützlich sein, wenn Sie die Parameter eines einzelnen Sterns sehen möchten.
- Die Dekonvolution-Funktionen von Siril unterstützen die Verwendung von dynamischen PSF-Messungen, um eine Dekonvolution-Funktion zu synthetisieren, die den direkt aus dem Bild gemessenen Sternparametern entspricht.

12.5 Konfiguration

Der dynamische PSF kann mit den Einstellungen im Dialogfeld Dynamischer PSF konfiguriert werden:

- Mit **Radius** wird die halbe Größe des Suchfelds festgelegt. Wenn Sie Probleme haben, bestimmte Sterne zu erkennen, können Sie versuchen, dies zu ändern, aber normalerweise ist die Standardeinstellung in Ordnung.
- **Schwellwert** ändert den Schwellenwert über dem Rauschen für die Sternerkennung. Wenn Sie diesen Wert erhöhen, werden weniger schwache Sterne erkannt. Bei sehr verrauschten Bildern sollten Sie dies dennoch tun. Wenn Sie diesen Wert verringern, werden zwar mehr schwache Sterne erkannt, aber der Algorithmus erkennt auch eher zufällige Helligkeitsspitzen im Rauschen als Sterne.
- Der Schwellenwert für die **Rundheit** legt die zulässige Elliptizität fest, ab der Sterne als solche erkannt werden. Stark elliptische Sterne können aufgrund von unvollkommener Nachführung oder Abbildungsfehlern auftreten, aber manchmal werden auch Doppelsterne, die zu nahe beieinander liegen, als ein einziger sehr länglicher Stern erkannt. Um all diese Probleme hervorzuheben, ist es möglich, eine höhere Grenze für die Rundheit festzulegen. Ein Maximalwert von 1 ist gleichbedeutend mit der Deaktivierung des Bereichs, so dass nur der Minimalwert übrig bleibt. Dieser Rundheitsbereich sollte für die Registrierung oder Astrometrie deaktiviert werden.
- **Umrechnung** legt ein Konvergenz-Kriterium fest, das vom Solver verwendet wird. Wird dieser Wert erhöht, kann der Solver mehr Interaktionen durchführen, um zu konvergieren, und möglicherweise zusätzliche Sterne entdecken, aber die Laufzeit des Solvers kann sich erhöhen.
- Der Profiltyp wählt zwischen **Gauß'schen** oder **Moffat'schen** Profilen für die Sterne aus.
- **Minimum beta** legt einen zulässigen Mindestwert für beta fest, damit eine Entdeckung als Stern akzeptiert wird. Galaxien können manchmal als Moffat-Profilsterne entdeckt werden, aber sie haben diffuse Profile und der Wert von Beta ist normalerweise sehr niedrig, weniger als etwa 1,5.
- **Relax PSF checks** ermöglicht eine Lockerung mehrerer Qualitätsprüfungen für Sternkandidaten. Dies wird wahrscheinlich zu einer signifikanten Zunahme von falsch-positiven Sternentdeckungen führen, oft mit wilden Parametern.
- Ein Bereich von **minimaler** und **maximaler Amplitude** kann eingestellt werden, um die Amplitude (Parameter mit der Bezeichnung `A` in den Berichten) der erkannten Sterne zu begrenzen. Dies ist nützlich, wenn nur nicht gesättigte Sterne ausgewählt werden sollen, z. B. für die PSF-Anpassung bei der Dekonvolution. Beachten Sie, dass das Entfernen der gesättigten Sterne aus der Erkennung die Registrierung und Astrometrie stören kann.

Tipp

Die in diesem Fenster definierten Einstellungen können an dem aktuell geladenen Bild getestet werden. Sie müssen jedoch bedenken, dass sie auch für alle Bilder der Sequenz verwendet werden, insbesondere für die Registrierungsmethode globale Sternausrichtung.

Der Befehl *findstar* befolgt die gleichen Einstellungen, die im Dialog Dynamischer PSF eingegeben wurden, kann aber auch mit dem Befehl *setfindstar* konfiguriert werden.

Siril Kommandozeile

```
findstar [-out=] [-layer=] [-maxstars=]
```

Erkennt Sterne im aktuell geladenen Bild, die einen höheren Pegel als den von Siril berechneten Schwellenwert haben.
Abschließend wird eine PSF angewendet, und Siril weist alle erkannten Strukturen zurück, die eine Reihe von vorgegebenen Erkennungskriterien nicht erfüllen, die mit dem Befehl SETFINDSTAR eingestellt werden können.
Schließlich wird eine Ellipse um die erkannten Sterne gezogen.

Der optionale Parameter **-out=** ermöglicht es, die Ergebnisse unter dem angegebenen Pfad zu speichern.
Die Option **-layer=** gibt die Ebene an, auf der die Erkennung durchgeführt wird (nur für Farbbilder).
Sie können auch die maximale Anzahl der erkannten Sterne begrenzen, indem Sie der Option **-maxstars=** einen Wert übergeben.

Siehe auch den Befehl CLEARSTAR

Verweise: *psf*, *setfindstar*, *clearstar*

Siril Kommandozeile

```
setfindstar [reset] [-radius=] [-sigma=] [-roundness=] [-focal=] [-pixelsize=] [-
↪convergence=] [ [-gaussian] | [-moffat] ] [-minbeta=] [-relax=on|off] [-minA=] [-
↪maxA=] [-maxR=]
```

Definiert die Parameter für die Sternerkennung für die Befehle FINDSTAR und REGISTER.

Wird kein Parameter übergeben, werden die aktuellen Werte aufgelistet.
Die Übergabe von **reset** setzt alle Werte auf die Standardwerte zurück. Sie können dann immer noch Werte nach diesem Schlüsselwort übergeben.

Konfigurierbare Werte:

-radius= definiert den Radius des anfänglichen Suchfeldes und muss zwischen 3 und 50 liegen.
-sigma= definiert die Schwelle über dem Rauschen und muss größer oder gleich 0,05 sein.

-roundness= definiert die minimale Rundheit der Sterne und muss zwischen 0 und 0,95 liegen. **-maxR** erlaubt es, eine Obergrenze für die Rundheit festzulegen, um nur die Bereiche zu visualisieren, in denen die Sterne deutlich elongiert sind. Nicht ändern für die Registrierung.

-minA und **-maxA** definieren Grenzwerte für die minimale und maximale Amplitude der Sterne, die zwischen 0 und 1 normiert sind.

-focal= definiert die Brennweite des Teleskops.

-pixelsize= definiert die Pixelgröße des Sensors.

-gaussian und **-moffat** konfigurieren das zu verwendende Solver-Modell (Gaussian ist der Standard).

Wenn Moffat gewählt wird, definiert **-minbeta=** den Mindestwert von beta, für den Kandidatensterne akzeptiert werden, und muss größer oder gleich 0,0 und kleiner als 10,0 sein.

-convergence= definiert die Anzahl der Iterationen, die zur Anpassung der PSF durchgeführt werden, und sollte zwischen 1 und 3 (toleranter) eingestellt werden.

-relax= entspannt die Prüfungen, die bei Sternkandidaten durchgeführt werden, um festzustellen, ob sie Sterne sind oder nicht, damit Objekte, die nicht wie Sterne geformt sind, trotzdem akzeptiert werden (standardmäßig ausgeschaltet)

Verweis: *findstar*, *register*, *psf*

12.6 Quellenverzeichnis

KAPITEL 13

FITS Header

Schlüsselwörter, die im Header einer FITS-Datei enthalten sind, können in Siril angezeigt werden. Klicken Sie dazu einfach auf *Werkzeuge→ FITS Header*.

Since version 1.3.0, it is possible to modify the value of keywords supported by Siril, either with the *update_key* command, or via the GUI, in the appropriate window. This window is divided into two tabs. The first, the *Header Editor*, lets you edit the keywords of the loaded image. When a sequence is loaded, the changes are applied to every image of the sequence (see the note below). It may vary from what the file header actually looks like, and represents more the state it will be in after saving. Keyword values are updated in real time. The second tab displays the header in text format as written in the file and as it was represented for Siril versions below 1.3.0.

Warnung

HISTORY-Schlüsselwörter werden nicht im Editor angezeigt. Sie sind auf der Registerkarte *Gespeicherter Header* sichtbar.

Eine andere Möglichkeit, den Header anzuzeigen, ist die Verwendung der Befehlszeile *dumpheader*. Dabei wird derselbe Header angezeigt, der auch auf der Registerkarte *Gespeicherter Header* angezeigt wird.

Warnung

Da SER-Dateien nur sehr wenige Schlüsselwörter enthalten und sich von FITS-Dateien unterscheiden, ist dieser Befehl auf diesen Sequenztyp nicht anwendbar.

Schlüsselname, Schlüsselwerte und Schlüsselkommentare können geändert werden, der Schlüssel muss jedoch ungeschützt sein. Zur einfachen Erkennung werden geschützte Schlüssel lachsfarben dargestellt.

Editing is very simple, just `double-click` on the cell to be modified. The first time selects the field, the second opens the edit mode. Pressing the `Enter` key validates the entry. To make the changes effective, remember to save the file.

Bemerkung

When a sequence is loaded, editing a key name, value or comment applies the change to the FITS header of **every image** of the sequence. A confirmation is asked the first time; editing then stays enabled until the window is closed. This is the graphical equivalent of the *sequpdate_key* command and is not available for SER sequences.

Warnung

Bitte beachten Sie, dass Siril die Gültigkeit des eingegebenen Werts nicht prüft. Es ist Sache des Benutzers, einen gültigen Wert einzugeben. Ein falscher Wert kann zu unerwünschtem Verhalten bei der Keyword-Verarbeitung von Siril führen.

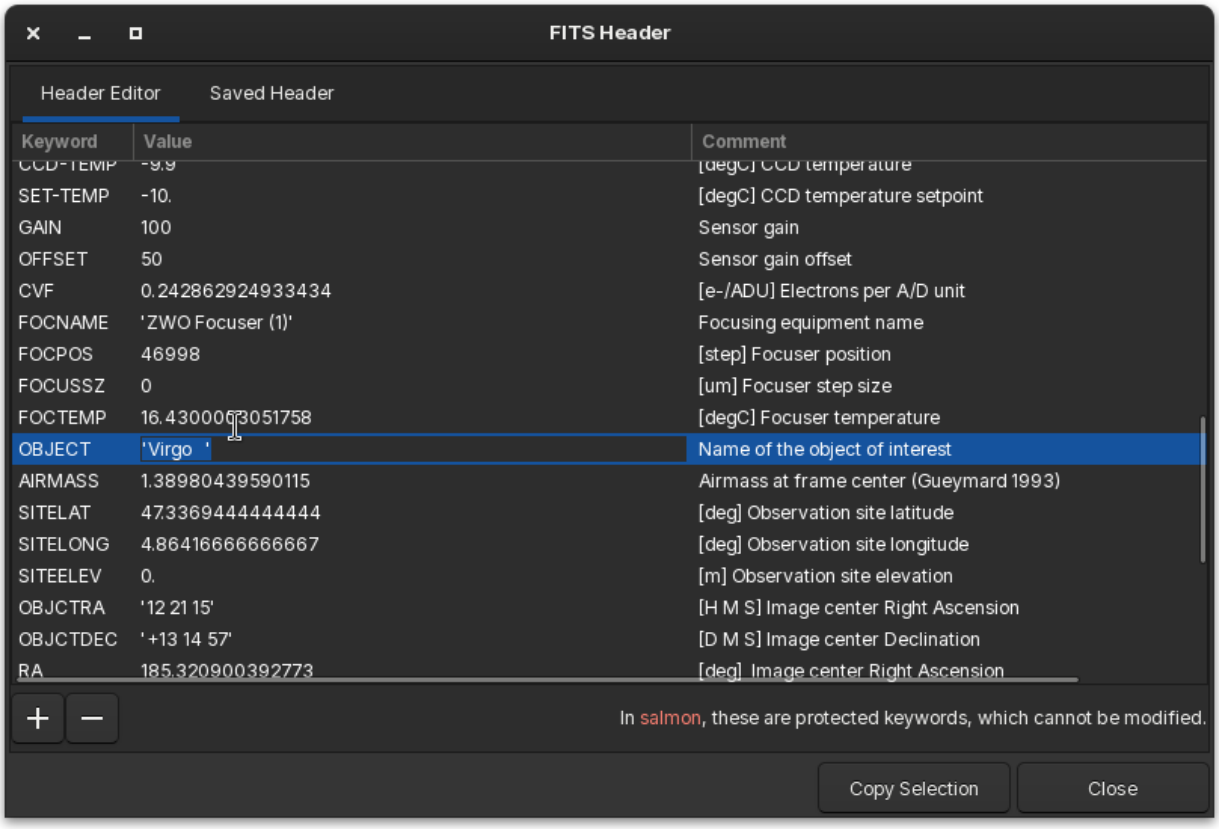


Abb. 1: FITS-Header-Dialogfeld beim Bearbeiten des Werts eines Schlüsselworts.

Das Fenster enthält eine Option in Form einer Schaltfläche *Auswahl kopieren*. Sie kopiert die ausgewählten Zeilen im Originalformat des FITS-Headers in die Zwischenablage und funktioniert für beide Registerkarten.

Schließlich können Sie mit der Schaltfläche + unten links ein neues Schlüsselwort hinzufügen und mit der Schaltfläche - ein bereits vorhandenes löschen:

1. Um ein Schlüsselwort hinzuzufügen, klicken Sie auf die Schaltfläche +. Ein neues Dialogfeld wird geöffnet, wie unten dargestellt. In diesem Fenster können Sie ein neues Schlüsselwort hinzufügen, dessen Name auf 8 Zeichen begrenzt ist (die Konvention `HIERARCH` wird nicht verwendet). Sie können das Feld *Kommentar* leer lassen, wenn die anderen beiden ausgefüllt sind. Der Benutzer kann auch nur das Feld *Kommentar* ausfüllen, um dem

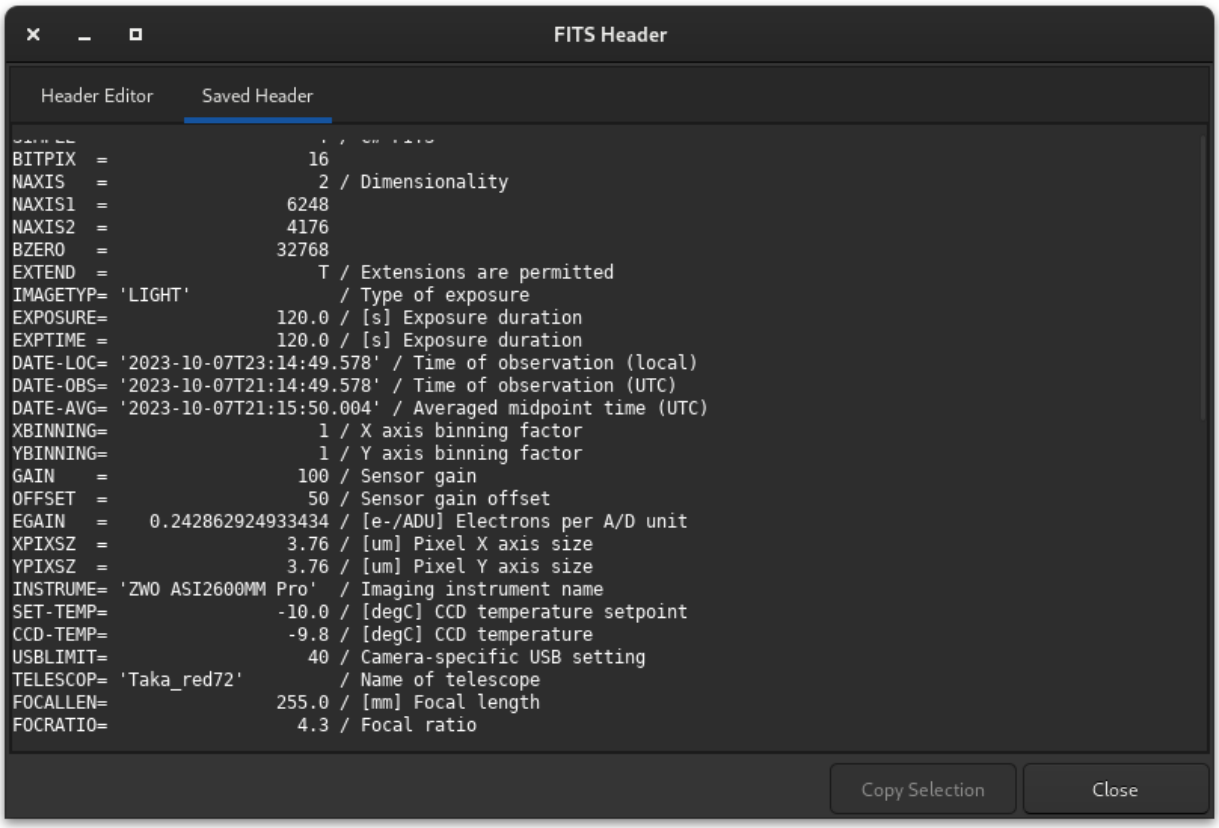


Abb. 2: Zweites Tab des FITS-Header-Dialogs.

FITS-Header einen einzelnen Kommentar hinzuzufügen. Wenn das Schlüsselwort vorhanden ist, wird sein Wert schließlich mit den bereitgestellten Informationen aktualisiert.

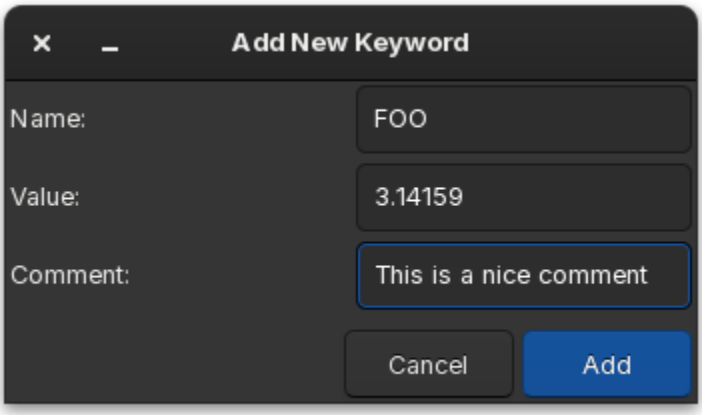


Abb. 3: Fenster zum Hinzufügen von Schlüsselworten

2. Um ein Schlüsselwort zu löschen, markieren Sie eine oder mehrere Zeilen und klicken Sie auf die Schaltfläche - (oder mit den Tasten `Entf` oder `Backspace`). Die markierten Schlüsselwörter werden gelöscht, sofern sie nicht geschützt sind.

> **Tipp**
>
> Beim Laden einer Sequenz ist es immer nur möglich, eine Zeile auf einmal auszuwählen. Das heißt, Sie können immer nur ein Schlagwort auf einmal aus einer Sequenz löschen. Für ein einzelnes Bild können Sie aber mehrere auswählen.

Siril Kommandozeile

```
dumpheader
```

Zeigt den FITS-Header-Wert für den angegebenen Schlüssel des geladenen Bildes in der Konsole an

Siril Kommandozeile

```
update_key key value [keycomment]
update_key -delete key
update_key -modify key newkey
update_key -comment comment
```

Aktualisiert das FITS-Schlüsselwort. Bitte beachten Sie, dass die Gültigkeit von **value** nicht überprüft wird. Diese Überprüfung liegt in der Verantwortung des Benutzers. Es ist auch möglich, einen Schlüssel mit der Option **-delete** vor dem Namen des zu löschenden Schlüssels zu löschen oder den Schlüssel mit der Option **-modify** zu ändern. Auf letztere muss der zu ändernde Schlüssel und der neue Schlüsselname folgen. Schließlich fügt die

Option **-comment**, gefolgt von Text, dem FITS-Header einen Kommentar hinzu. Bitte beachten Sie, dass jeder Text, der Leerzeichen enthält, in doppelte Anführungszeichen eingeschlossen werden muss

Siril Kommandozeile

```
sequpdate_key sequencename key value [keycomment]
sequpdate_key sequencename -delete key
sequpdate_key sequencename -modify key newkey
sequpdate_key sequencename -comment comment
```

Gleicher Befehl wie UPDATE_KEY, aber für die Sequenz **Sequenzname**. Dieser Befehl funktioniert jedoch nicht bei SER-Sequenzen

Links: *update_key*

KAPITEL 14

Intensitätsprofilierung

Siril verfügt über einen Intensitätsprofilierungsmodus. Der Benutzer wählt eine Linie zwischen zwei Punkten aus und Siril erstellt ein Diagramm der Pixelwerte zwischen diesen Punkten. Dies hat mehrere Anwendungsmöglichkeiten. Es kann zur Untersuchung des Intensitätsprofils eines einzelnen Sterns, zur Erstellung eines Profils einer ganzen Galaxie verwendet werden.

14.1 Basis-Intensitätsprofil

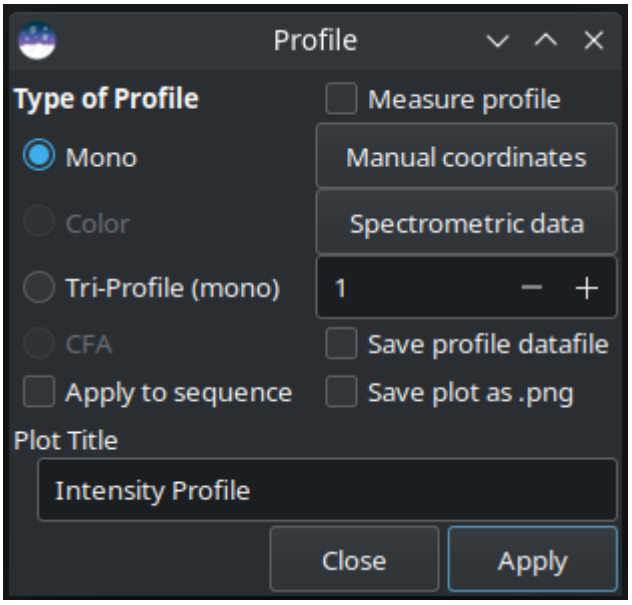
Um ein Basis-Intensitätsprofil eines Sterns oder eines anderen Objekts zu erstellen, wählen Sie die Schaltfläche Profil in der unteren Symbolleiste. Dadurch wird Siril in den Profilierungsmodus versetzt und ein kleiner Dialog geöffnet.

Sie können nun auf die Hauptbildanzeige klicken und ziehen, um den Anfangs- und Endpunkt der Linie festzulegen, die Sie profilieren möchten. Wenn Sie beim Ziehen der Linie die `Umschalttaste` gedrückt halten, rastet die Linie entweder horizontal oder vertikal ein.

Tipp

Wenn die Profillinie genau horizontal oder genau vertikal verläuft, können exakte Pixelwerte direkt aus dem Bild verwendet werden. Wenn die Profillinie weder horizontal noch vertikal ist, fallen die zu zeichnenden Punkte nicht genau auf ein Pixel und es werden daher bilinear interpolierte Pixelwerte verwendet.

Ein benutzerdefinierter Titel für Ihr Diagramm kann in das Steuerelement am unteren Rand des Dialogs eingegeben werden.



Tipp

Bei der Verarbeitung einer Sequenz ist es möglich, dass der benutzerdefinierte Titel die Bildnummer und die Gesamtzahl anzeigt, indem Sie () an das Ende des Titels anhängen. Wenn Sie z. B. **Solarspektren** () als Titel für eine 5-Bild-Sequenz eingeben, werden die Titel **Solarspektren (1 / 5)**, **Solarspektren (2 / 5)** usw. erzeugt. Die Klammern werden ignoriert und entfernt, wenn ein einzelnes Bild verarbeitet wird.

14.2 Profiltypen

Verwenden Sie die Optionsfelder, um den gewünschten Profiltyp auszuwählen. (Klicken Sie auf die Beispielbilder unten, um sie in voller Größe zu sehen).

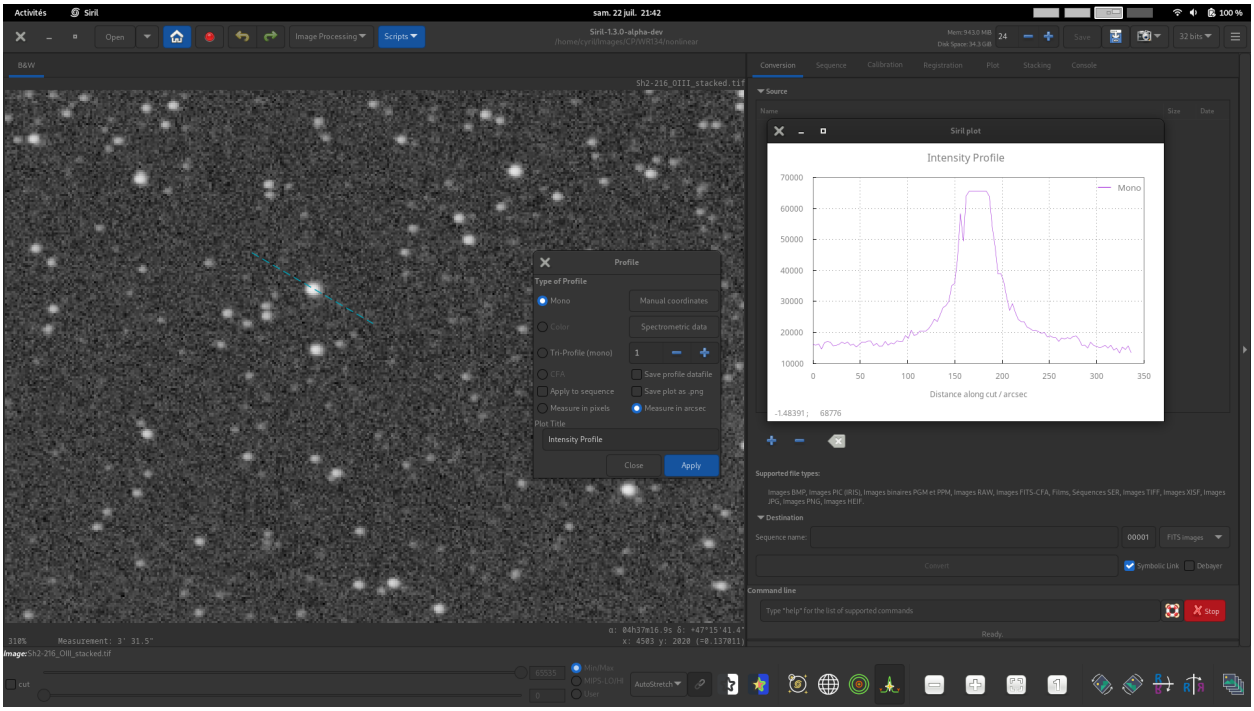
- **Mono-Profil**. Für Mono- oder Farbbilder wird ein Leuchtdichteprofil zwischen zwei Punkten erstellt. Dieser Modus kann mit spektrometrischen Daten verwendet werden.

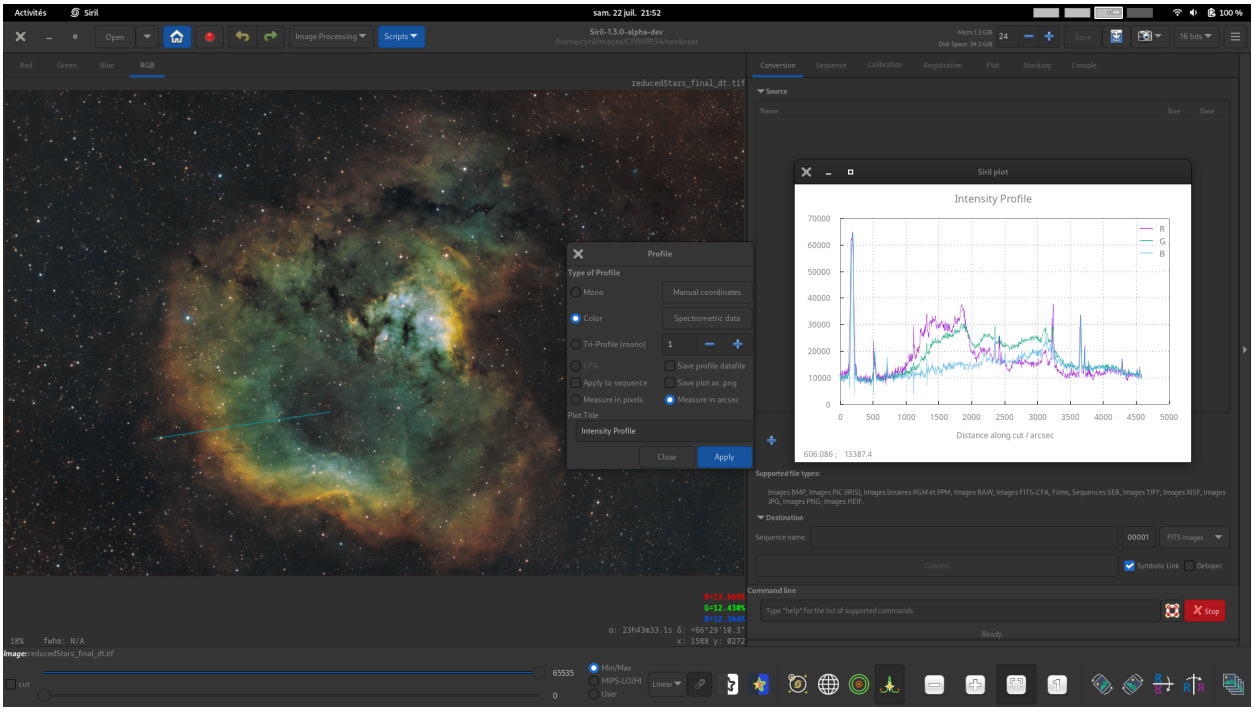
Tipp

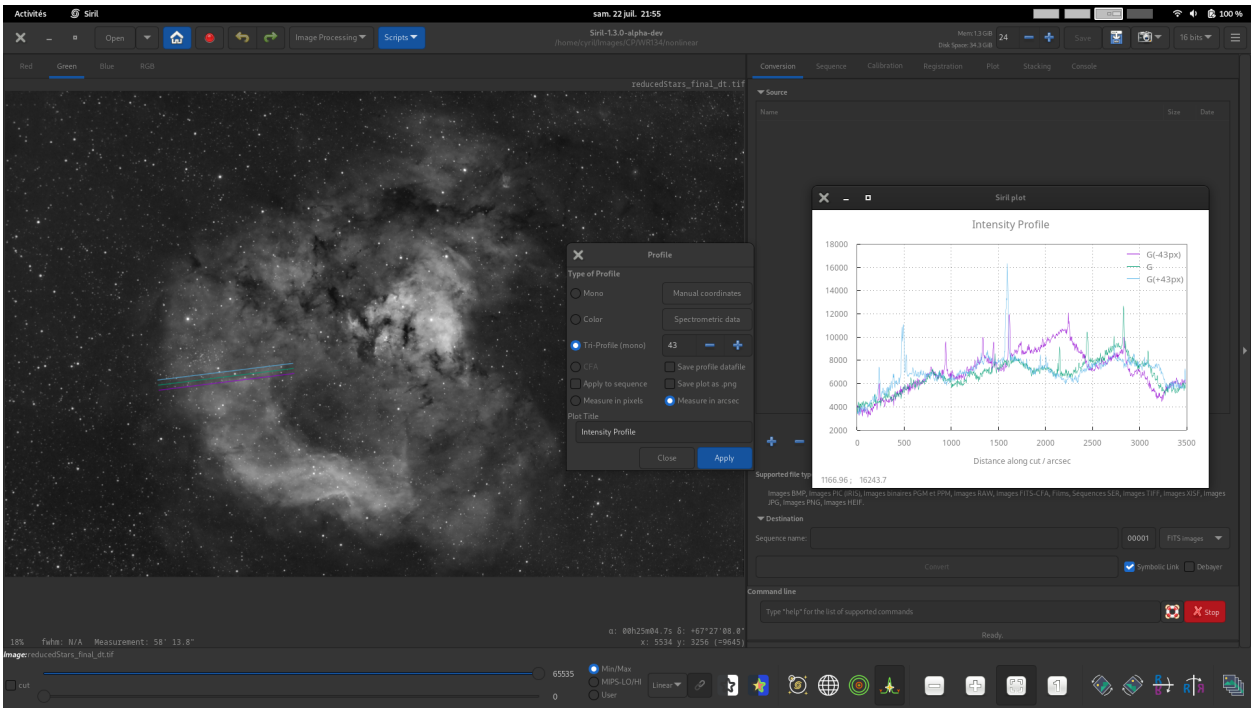
Wenn ein Farbbild geladen ist, aber der Mono-Profilierungsmodus ausgewählt ist, wird das Profil entsprechend dem Ansichtsfenster erstellt. Die Ansichtsfenster R, G und B liefern Monoprofile für den jeweiligen Kanal und das Ansichtsfenster RGB liefert ein Luminanzprofil, bei dem alle drei Kanäle gleich gewichtet werden.

- **Farbprofil**. Für Farbbilder werden drei Profile für die R-, G- und B-Pixelwerte zwischen zwei Punkten erstellt. Dieser Modus kann mit spektrometrischen Daten verwendet werden.
- **Tri-Profil (Mono)**. Für Mono- oder Farbbilder werden drei parallele, gleichabständige Leuchtdichteprofile zwischen zwei Punkten erzeugt. Der Abstand zwischen den 3 Profilen kann über die Schaltfläche "Drehen" eingestellt werden.
- **CFA**. Bei Bildern, die nur ein Bayer-Muster aufweisen, werden vier Profile für die vier CFA-Unterkanäle zwischen zwei Punkten erstellt. Dies kann besonders nützlich sein, um das Profil von Flats mit Bayer-Muster oder anderen Bildern mit Bayer-Muster zu prüfen, bevor sie debayered werden.

Klicken Sie auf *Anwenden*, um Ihr Profil zu erstellen.


Activités
Siril
sam. 22 juil. 21:42
Open
Image Processing
Scripts
Siril-1.3.0-alpha-dev
/home/cyril/Images/CP/WR134/nonlinear
B&W
Sh2-216_OIII_stacked.tif
Profile
Type of Profile
Mono
Manual coordinates
Color
Spectrometric data
Tri-Profile (mono)
CFA
Save profile datafile
Apply to sequence
Save plot as .png
Measure in pixels
Measure in arcsec
Plot Title
Intensity Profile
Close
Apply
Conversion
Sequence
Calibration
Registration
Plot
Stacking
Console
Source
Siril plot
Intensity Profile
Mono
Distance along cut / arcsec
Supported file types:
Destination
Sequence name:
Convert
Symbolic Link
Debayer
Command line
Type "help" for the list of supported commands
Stop
Ready.
Measurement: 3' 31.5"
Image: Sh2-216_OIII_stacked.tif
AutoStretch


Activités
Siril
sam. 22 juil. 21:52
Open
Image Processing
Scripts
Siril-1.3.0-alpha-dev
/home/cyril/Images/CP/WR134/nonlinear
Red
Green
Blue
RGB
reducedStars_final_dt.tif
Profile
Type of Profile
Mono
Manual coordinates
Color
Spectrometric data
Tri-Profile (mono)
CFA
Save profile datafile
Apply to sequence
Save plot as .png
Measure in pixels
Measure in arcsec
Plot Title
Intensity Profile
Close
Apply
Conversion
Sequence
Calibration
Registration
Plot
Stacking
Console
Source
Siril plot
Intensity Profile
R
G
B
Distance along cut / arcsec
Supported file types:
Destination
Sequence name:
Convert
Symbolic Link
Debayer
Command line
Type "help" for the list of supported commands
Stop
Ready.
Image: reducedStars_final_dt.tif
Linear



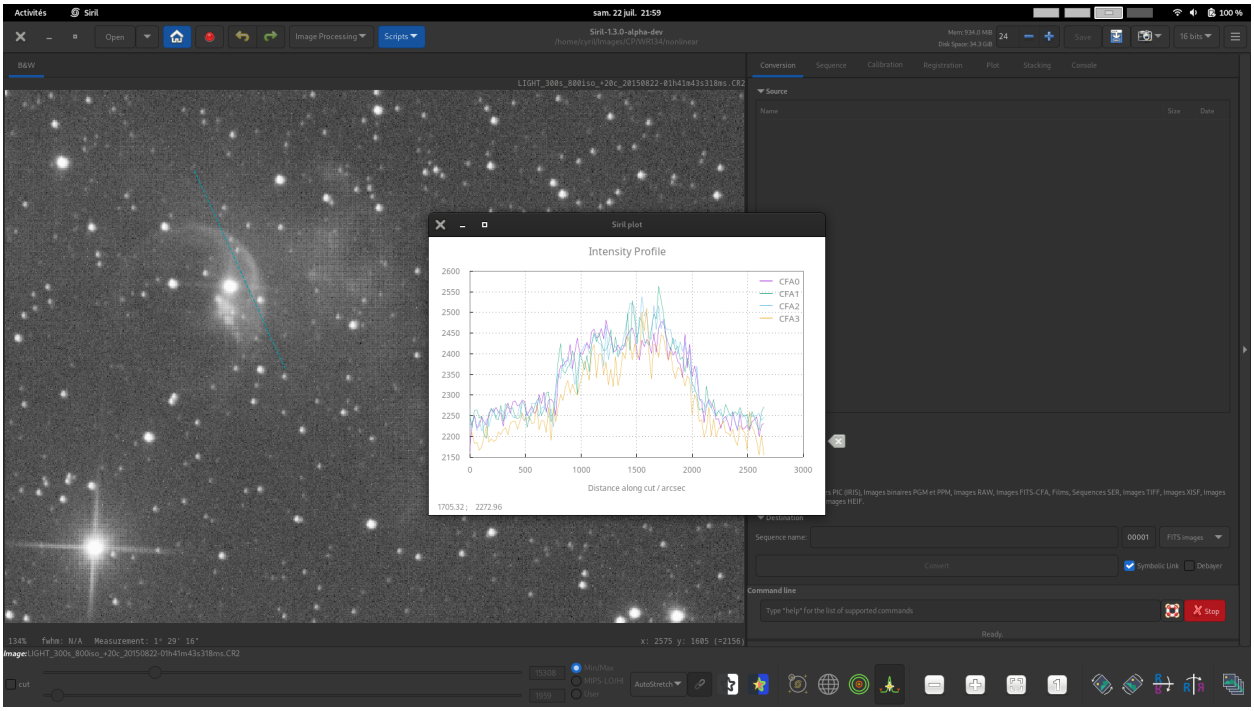


Abb. 1: Dieses Bild zeigt die Verwendung des Steuerelements Benutzerdefinierter Titel zum Festlegen eines benutzerdefinierten Titels für die Darstellung.

14.3 Präzise Koordinateneingabe

Um die präzise und wiederholbare Eingabe von Koordinaten zu erleichtern, gibt es eine manuelle Eingabemethode. Klicken Sie auf die Schaltfläche *Manuelle Koordinaten* und Sie können die X- und Y-Koordinaten des Start- und Endpunkts der Profillinie eingeben. Wenn eine Profillinie bereits gezeichnet ist, aber ein Punkt nicht ganz an der gewünschten Stelle liegt, können Sie diesen Popup-Dialog verwenden, um die Platzierung der Endpunkte fein abzustimmen.

If you wish to set an endpoint exactly to the position of a star, make a rectangular selection around the star and click the relevant star button to the right of the dialog.

> **Bemerkung**
>
> Bei Verwendung des CFA-Modus werden die Koordinaten im *Eingabebild* angegeben. Jeder CFA-Kanal ist jedoch halb so breit und halb so hoch. Die x-Achse in der Darstellung im CFA-Modus wird in Pixeln im CFA-Unterkanal gemessen, d. h. sie umfasst die halbe Anzahl von Pixeln wie im Eingabebild.

14.4 Messungen

Die Intensitätsprofillinie kann auf zwei Arten als Messinstrument verwendet werden:

- Wenn Sie das Kontrollkästchen *Profil messen* aktivieren, werden alle mit der Maus gezogenen Profillinien gemessen, ähnlich wie bei der Schnellmessfunktion `Strg` + `Umschalt` + `Ziehen`.
- Im Dialogfeld Koordinaten gibt es eine Schaltfläche *Messen*. Diese bietet die gleiche Messfunktion, ermöglicht es Ihnen aber, die Endpunkte genau zu bestimmen und die Profillinie dann bei Bedarf zu messen. Durch die Auswahl von Sternen, Kleinplaneten oder Kometenkernen als Startpunkt, wie oben beschrieben, können Messungen zwischen zwei Himmelskörpern sehr präzise (mit Sub-Pixel-Präzision) durchgeführt werden.

> **Bemerkung**
>
> Die Siril-Messfunktion verwendet die Kleinwinkel-Approximation für den Winkelabstand θ. Der größte Fehlerterm ist proportional zu θ^3 und beträgt weniger als 1% für Messungen bis zu 10°: Er ist daher für die meisten astrometrischen Anwendungen gültig, wird aber bei großen Messungen über Ultra-Weitwinkel-Bilder ungenau. Bei Messungen über 10° wird eine Warnung in das Protokoll geschrieben.

14.5 Siril Grafikwerkzeug

Die Profilerstellungsfunktion verwendet das Siril-interne Grafikwerkzeug zur Darstellung der verschiedenen Profile. Mit den erzeugten `*.dat`-Dateien können Sie weiterhin jedes beliebige Plot-Tool verwenden, um die zugrunde liegenden Daten zu untersuchen.

Ein `Rechtsklick` auf eine beliebige Stelle der Plotfläche öffnet ein Kontextmenü:

- Gitter und Legende anzeigen/verbergen
- Aktuelle Ansicht in Zwischenablage exportieren, `*.png` or `*.svg`
- Zugrunde liegende Daten in einer Datei `*.dat` speichern

Beachten Sie, dass alle Exporte den aktuellen Zoom/Ausschnitt berücksichtigen, während beim Speichern in dat die Daten ungefiltert exportiert werden.

Die folgenden GUI-Interaktionen stehen zur Verfügung:

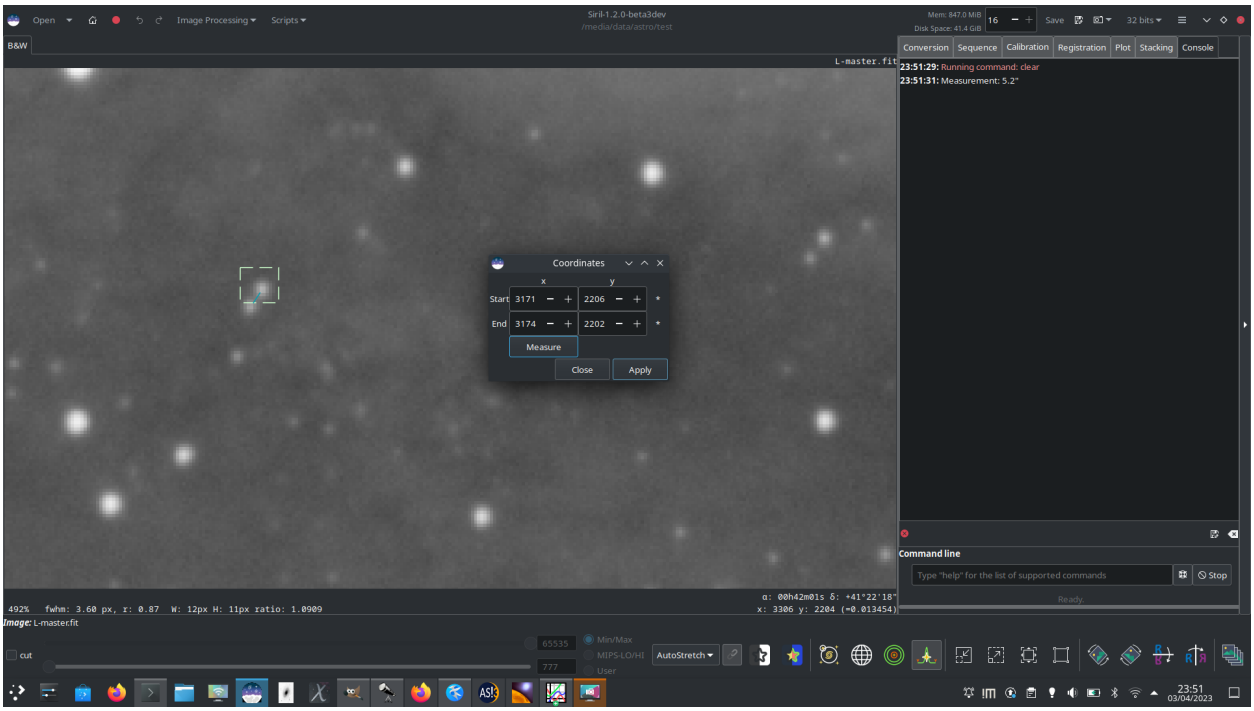


Abb. 2: Hier wurden zwei nahe Sterne ausgewählt und als Endpunkte festgelegt, wobei der Abstand zwischen ihnen mit 5,2 Bogensekunden gemessen wurde. Dies könnte zur Untersuchung naher Doppelsterne oder zur Triangulation der Position eines Kleinplaneten verwendet werden.

- `Klicken` + `Ziehen`, um eine Auswahl zu treffen. Der Zoom wird auf den ausgewählten Bereich gesetzt, wenn die Maus losgelassen wird.
- `Strg` + `Ziehen` zum Verschieben der aktuellen Ansicht.
- `Strg` + `Scrollrad` zum Vergrößern/Verkleinern.
- `Doppelklick` zum Zurücksetzen auf die Standardposition/den Standardzoom.

14.6 Befehle

Siril Kommandozeile

```
profile -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] { [-savedat] | [-filename=] } [-
↪layer=] [-width=] [-spacing=] ["-title=My Plot"]
```

Erzeugt ein Intensitätsprofil zwischen 2 Punkten im Bild, auch bekannt als *Schnitt*. Die Argumente können in beliebiger Reihenfolge angegeben werden. Die Argumente **-to=x,y** und **-from=x,y** sind obligatorisch.

Das Argument **-layer={red | green | blue | lum | col}** gibt an, welcher Kanal (oder Luminanz oder Farbe) gezeichnet werden soll, wenn das Bild farbig ist. Es kann auch in Verbindung mit der Option **-tri** verwendet werden, die drei parallele, gleichabständige Profile erzeugt, die jeweils durch **-spacing=** Pixel voneinander getrennt sind. Beachten Sie jedoch, dass bei Tri-Profilen die Option **col** genauso behandelt wird wie **lum**.

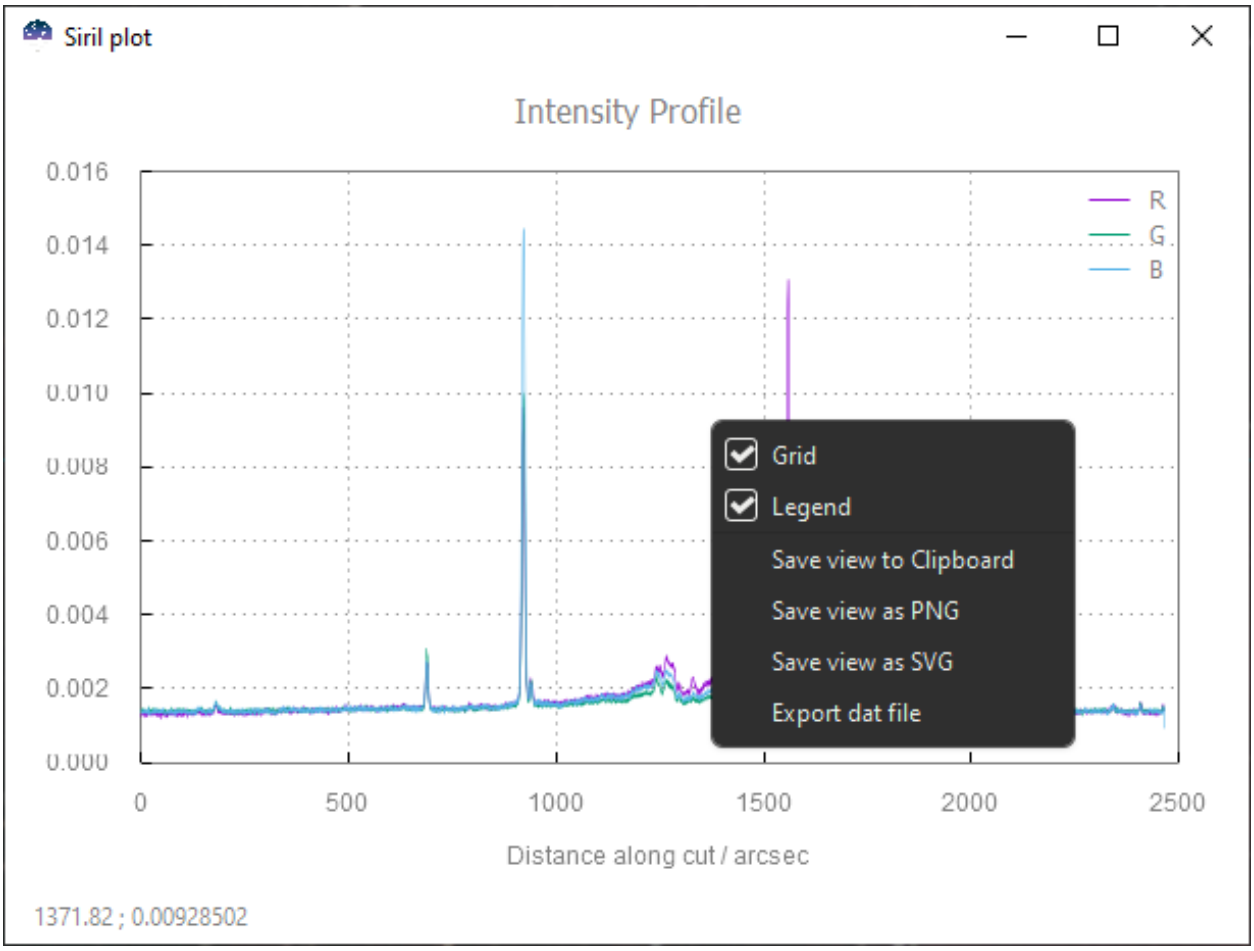


Abb. 3: Kontextmenü für Siril-Grafiken

Mit der Option **-cfa** wird der CFA-Modus ausgewählt, der 4 Profile erzeugt: 1 für jeden CFA-Kanal in einem Bild mit Bayer-Muster. Diese Option kann nicht mit Farbbildern oder Schwarzweißbildern ohne Bayer-Muster verwendet werden und kann nicht gleichzeitig mit der Option **-tri** verwendet werden.

Die Option **-arcsec** bewirkt, dass die x-Achse die Entfernung in arcsec anzeigt, wenn die erforderlichen Metadaten verfügbar sind. Wenn die Option nicht angegeben ist oder keine Metadaten vorhanden sind, wird die Entfernung in Pixel-Einheiten angezeigt.

Das Argument **-savedat** bewirkt, dass die Datendateien gespeichert werden: Der Dateiname wird in das Protokoll geschrieben. Alternativ kann das Argument **-filename=** verwendet werden, um einen Dateinamen anzugeben, in den die Datendatei geschrieben werden soll. (Die Option **-filename=** impliziert **-savedat**.)

Das Argument **"-title=Mein Titel "** setzt einen benutzerdefinierten Titel "Mein Titel"

Siril Kommandozeile

```
seqprofile sequence -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] [-savedat] [-layer=] [-
↪width=] [-spacing=] [ {-xaxis=wavelength | -xaxis=wavenumber } ] [{-wavenumber1= | -
↪wavelength1=} -wn1at=x,y {-wavenumber2= | -wavelength2=} -wn2at=x,y] ["-title=My␣
↪Plot"]
```

Erstellt ein Intensitätsprofildiagramm zwischen zwei Punkten in jedem Bild der Sequenz. Nach dem obligatorischen ersten Argument, das die zu verarbeitende Sequenz angibt, sind die anderen Argumente die gleichen wie für den Befehl **profile**. Wenn eine Sequenz verarbeitet wird und gewünscht wird, dass die aktuelle Bildnummer und die Gesamtzahl der Bilder im Format „Meine Sequenz (1 / 5)“ angezeigt werden, sollte der angegebene Titel mit () enden (z. B. „Meine Sequenz ()“ und die Nummern werden automatisch ausgefüllt)

KAPITEL 15

Bildinspektoren

Siril verfügt über mehrere Werkzeuge, die Ihnen bei der Analyse Ihres Bildes helfen und Ihnen Informationen zur Qualität der Aufnahme geben können. Insbesondere, ob Ihr Setup optische Defekte hat oder nicht.

15.1 Verkippung

Das erste von Siril vorgeschlagene Werkzeug ist die Verkippungsanalyse. Eine Verkippung tritt auf, wenn der Sensor nicht orthogonal zur Abbildungsebene steht: Dies erfordert einen Eingriff in das optische System. Sie können diese Funktion auf zwei verschiedene Arten ausführen. Entweder klicken sie in der grafischen Benutzeroberfläche auf die Verkippung-Schaltfläche (im Menü *Werkzeuge* → *Bildanalyse* → *Verkippung anzeigen*) oder über die Kommandozeile. Letzteres bietet sogar die Möglichkeit die Verkippung über eine komplette Sequenz zu ermitteln, um eine größere Genauigkeit zu erreichen. Der folgende Befehl:

Siril Kommandozeile

```
tilt [clear]
```

Berechnet die Verkippung des Sensors als FWHM-Differenz zwischen dem besten und dem schlechtesten abgeschnittenen Mittelwert der Ecken. Die Option **clear** erlaubt es, die Darstellung zu löschen

gibt Folgendes aus:

```
22:28:13: Running command: tilt
22:28:13: Findstar: processing for channel 0...
22:28:15: Stars: 7598, Truncated mean[FWHM]: 3.40, Sensor tilt[FWHM]: 0.31 (9%), Off-
↪axis aberration[FWHM]: 0.39
```

In der Konsole werden angezeigt:

- die Anzahl der für die Messung verwendeten Sterne

- der durchschnittliche FWHM-Wert auf dem Sensor, frei von Ausreißern
- die Verkippung, ausgedrückt als Differenz zwischen der besten und der schlechtesten FWHM an den vier Ecken des Bildes mit dem Prozentsatz der Neigungsabweichung in Klammern (ein Wert von mehr als 10% weist auf ein Neigungsproblem hin)
- die Bildfeldwölbung (Abberation), ausgedrückt durch die Differenz der FWHM zwischen den Sternen in der Mitte und den Sternen am Rand des Sensors

Wenn die Anzahl der erkannten Sterne klein ist (<200), ermöglichen die Sternerkennungseinstellungen der dynamischen PSF eine Verbesserung durch Anpassung des Schwellenwertes/Radius. Je größer die Anzahl der in die Berechnung einbezogenen Sterne ist, desto zuverlässiger ist das Ergebnis der Analyse.

Warnung

Damit das Ergebnis sinnvoll ist, sollte dieser Befehl vorzugsweise mit einem Einzelbild und nicht mit einem Summenbild ausgeführt werden. Ein kalibriertes Einzel-Lights (für Farbsensoren nur debayered) ist daher ideal. Außerdem sind die Proportionen des gezeichneten Vierecks übertrieben, um auf dem Bildschirm besser sichtbar zu sein. Es kann nicht genau der Realität entsprechen.

Abb. 1: Anzeige des Verkippungsdiagramms

Tipp

Neben der Verwendung des Befehls "tilt -clear" kann das Tilt-Diagramm auch über die Schaltfläche *Entfernen* im

Dialogfeld Dynamische PSF gelöscht werden.

Siril Kommandozeile

```
seqtilt sequencename
```

Derselbe Befehl wie TILT, aber für die Sequenz **sequencename**. Ergibt generell bessere Resultate

Verweit: *tilt*

15.2 Inspektor für die Bildfeldwölbung (Abberration Inspektor)

Dieses Werkzeug erstellt ein 3x3-Mosaik aus der Bildmitte, den Ecken und Rändern. So lässt sich die Form der Sterne in verschiedenen Bildbereichen leicht vergleichen. Sie können auf diese Funktion zugreifen, indem Sie mit der rechten Maustaste auf das Bild klicken und *Werkzeuge→ Bildanalyse → Abberationsprüfung* auswählen. Sie können die Einstellungen dieses Werkzeugs, wie z.B. die Größe der Felder und des Fensters, in den *Einstellungen* einstellen.

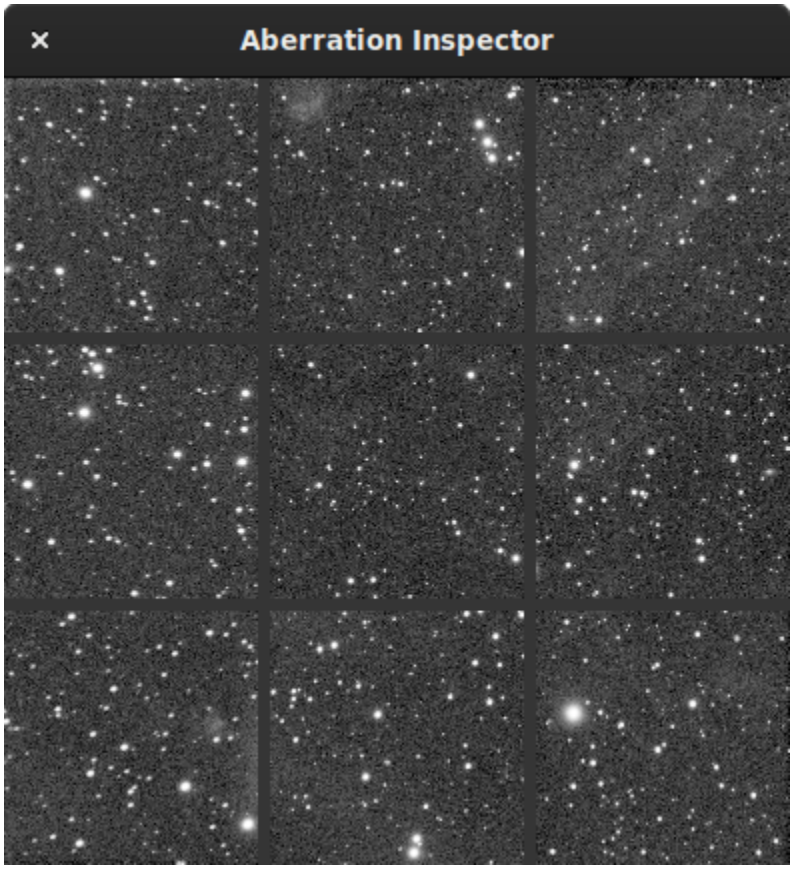


Abb. 2: Das Fenster des Bildfeldwölbungs-Inspektors zeigt die durch das optische System bedingten Verzerrungen der Sterne in den Randbereichen des Bildes.

Es ist auch ein sehr gutes Hilfsmittel, um festzustellen, ob das Bild einen Farbverlauf/Gradienten enthält: Die Unter-

schiede in Helligkeit und/oder Farbe werden deutlich sichtbar.

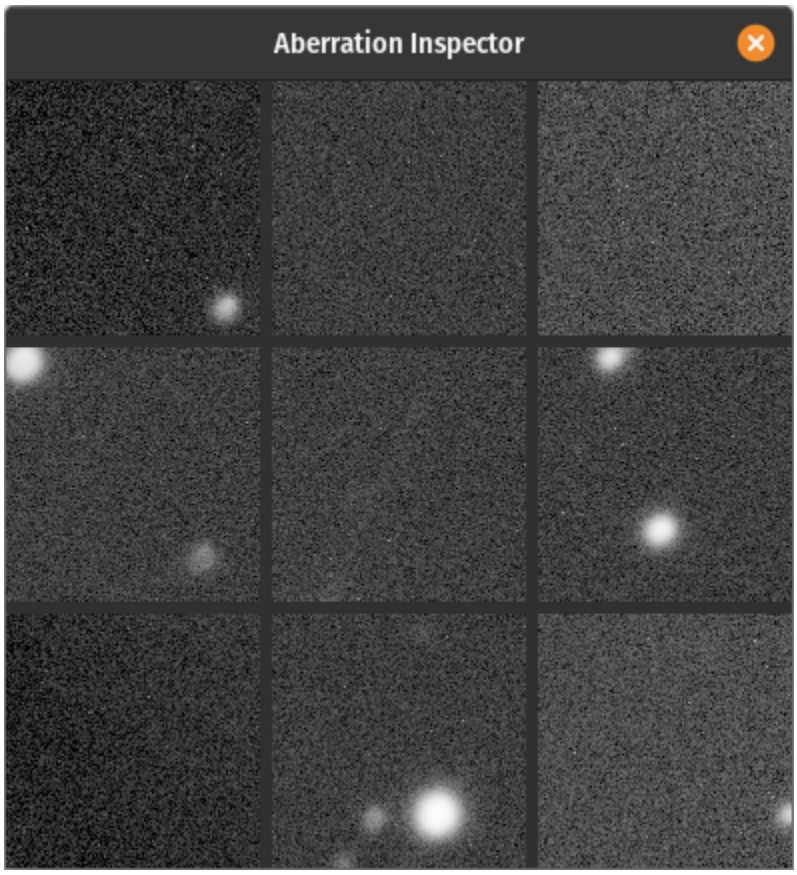


Abb. 3: Das Fenster des Bildfeldwölbungs-Inspektors zeigt Helligkeitsunterschiede an.

Siril Kommandozeile

```
inspector
```

Teilt das aktuelle Bild in ein Neun-Panel-Mosaik, das die Bildecken und die Mitte für eine genauere Betrachtung zeigt (nur GUI/Benutzeroberfläche)

KAPITEL 16

Photometrie

In diesem Abschnitt werden Sie mit allen Hilfsmitteln im Zusammenhang mit der Photometrie vertraut gemacht. Zunächst werden die Grundsätze der Photometrie erläutert, dann wird erklärt, wie sie in Siril verwendet werden.

Siril ist in der Lage, die Helligkeit von Sternen sowie die Ungenauigkeit der Helligkeit zu bestimmen. Von dieser Information aus ist es möglich, Helligkeitsveränderungen bestimmter Sterne, Exoplaneten oder Bedeckungen zu untersuchen. Am Ende des Prozesses wird auch eine Lichtkurve erstellt.

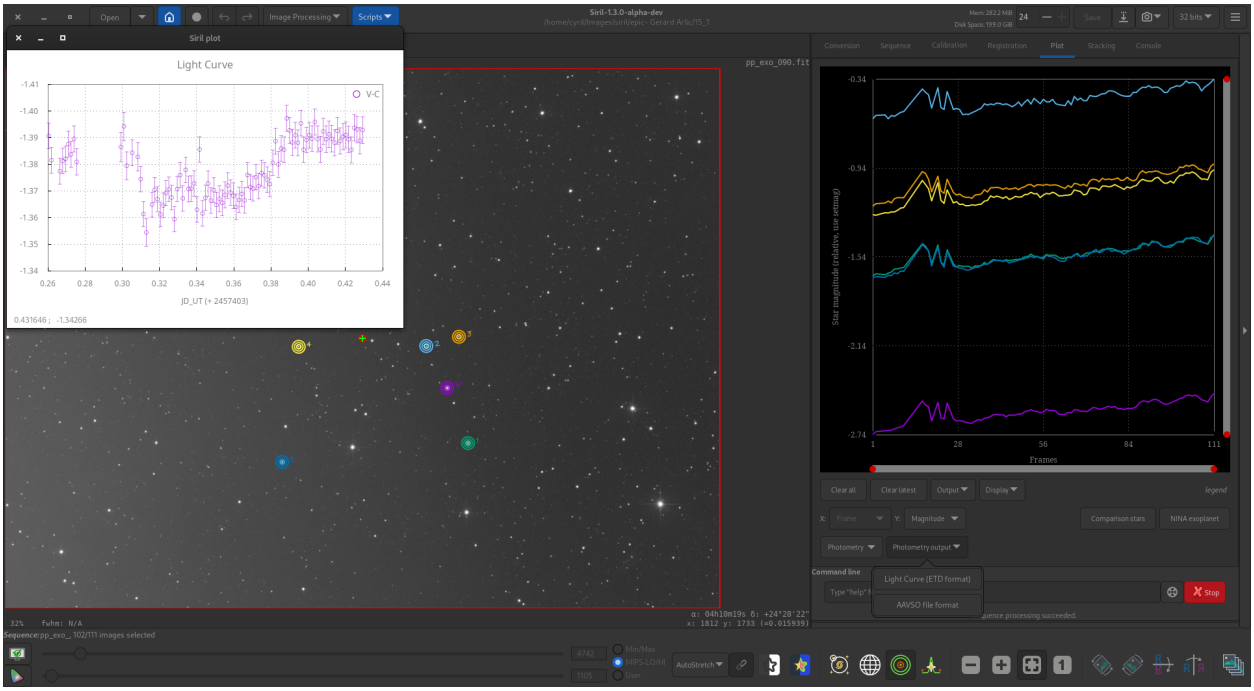


Abb. 1: Beispiel einer Exoplaneten-Photometrie in Siril.

16.1 Grundlagen

Die Photometrie ist die Wissenschaft von der Messung des Lichts. Sie zielt darauf ab, den Fluss oder die Intensität des von astronomischen Objekten ausgestrahlten Lichts zu messen. In Siril kann die Photometrie verwendet werden, um die Lichtkurve veränderlicher Sterne, Transits von Exoplaneten oder Sternbedeckungen zu analysieren oder um Farben in RGB-Bildern zu kalibrieren.

Die verwendete Methode ist die Aperturphotometrie. Ihr Grundprinzip besteht darin, den beobachteten Lichtstrom in einem bestimmten Radius um das Zentrum eines Objekts zu summieren und dann den Gesamtbeitrag des Himmelshintergrunds in derselben Region abzuziehen (berechnet im Ring zwischen dem inneren und dem äußeren Radius, unter Ausschluss der abweichenden Pixel), so dass nur der Lichtstrom des Objekts übrig bleibt, um eine instrumentelle Helligkeit zu berechnen. Dies ist in der folgenden Abbildung dargestellt.

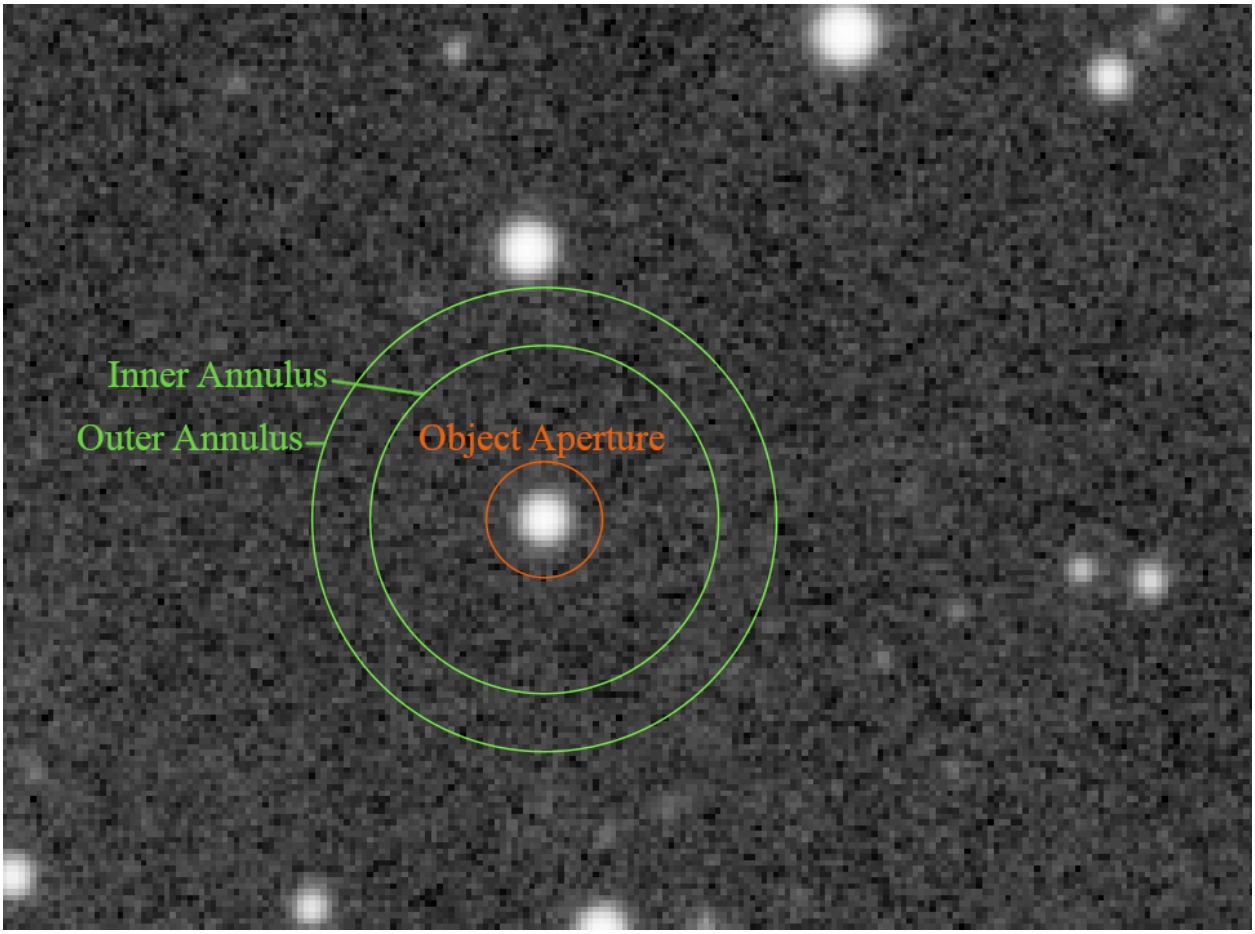


Abb. 2: Kreise der Aperturphotometrie

Die Werte dieser Einstellungen können in der Sektion *Photometrie* der Voreinstellungen geändert werden. *aperture* muss alle Pixel des zu messenden Objekts enthalten, *annulus* sollte dagegen keine seiner Pixel enthalten. Standardmäßig wird *aperture* für ein Ziel mit der doppelten FWHM der PSF eingestellt, aber die Größe des Annulus ist fest. Diese Werte sollten für ein bestimmtes Sampling angepasst und sorgfältig überprüft werden.

Bemerkung

Der folgende Text ist eine gekürzte und modifizierte Kopie der ausgezeichneten MuniPack-Softwaredokumentation von David Motl, die unter der GNU Free Documentation License veröffentlicht wurde und deren Quellen hier

verfügbar sind.

Messung der Helligkeit eines Objekts

Die Summe S der Pixel in einem kleinen Bereich A um ein Objekt ist die Summe der Nettointensität I des Objekts plus der Hintergrundintensität $B \cdot A$:

$$S = I + B \cdot A \tag{16.1}$$

Die Werte von S und B werden aus dem Quellbild abgeleitet, die Fläche A wird als Fläche eines Kreises mit dem Radius r bestimmt, wobei r die Größe der Blende in Pixeln ist. Es ist dann einfach, die Nettointensität I eines Objekts in ADU zu berechnen:

$$I = S - B \cdot A \tag{16.2}$$

Unter der Annahme, dass die Nettointensität I proportional zum beobachteten Fluss F ist, können wir die scheinbare Helligkeit m des Objekts mit Hilfe des Pogsonschen Gesetzes ableiten:

$$m = -2.5 \log_{10} \left(\frac{I}{I_0} \right) \tag{16.3}$$

Abschätzung des Messfehlers

Nachdem wir die instrumentelle Rohhelligkeit eines Objekts ermittelt haben, werden wir versuchen, seinen Standardfehler zu schätzen. Zunächst werden wir uns an einige allgemeine Regeln erinnern, die für den Standardfehler und seine Ausbreitung gelten. Dies ist eine allgemeine Regel für die Fehlerfortpflanzung durch eine Funktion f mit unsicherem Wert X:

$$\mathrm{Var}(f(X)) = \left(\frac{df}{dx} \right)^2 \mathrm{Var}(X) \tag{16.4}$$

Aus dieser allgemeinen Regel leiten wir zwei Gesetze der Fehlerfortpflanzung ab. Im ersten Fall wird der unsichere Wert X mit einer Konstanten a multipliziert und um einen konstanten Offset b verschoben. Dieses Gesetz kann auch für den Fall angewendet werden, dass nur eine Multiplikation oder nur eine Verschiebung stattfindet.

$$\mathrm{Var}(aX + b) = a^2 \, \mathrm{Var}(X) \tag{16.5}$$

Das zweite Gesetz definiert den Fehler eines Logarithmus einer unbestimmten Größe X:

$$\mathrm{Var}(\log(\pm bX)) = \frac{\mathrm{Var}(X)}{\bar{X}^2} \tag{16.6}$$

Bitte beachten Sie, dass die *log*-Funktion hier der natürliche Logarithmus ist, während die Pogson-Formel (siehe oben) den Logarithmus zur Basis 10 enthält. Die folgende Gleichung hilft uns, mit diesem Unterschied umzugehen:

$$\log_b(x) = \frac{\log_k(x)}{\log_k(b)} \tag{16.7}$$

Setzt man diese beiden Gleichungen zusammen, erhält man:

$$\mathrm{Var}(\log_{10}(\pm bX)) = \frac{\mathrm{Var}(X)}{\bar{X}^2 \, \log(10)^2} \tag{16.8}$$

Wenn wir zwei unkorrelierte unbestimmte Variablen X und Y haben, ist die Varianz ihrer Summe die Summe ihrer Varianzen, diese Gleichung ist als Bienaymé-Formel bekannt.

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) \tag{16.9}$$

Aus dieser Formel lässt sich auch der Standardfehler eines Stichprobenmittelwerts ableiten. Wenn wir N Beobachtungen einer Zufallsvariablen X mit einer stichprobenbasierten Schätzung des Standardfehlers der Grundgesamtheit s haben, dann ist der Standardfehler einer Stichprobenmittelwertschätzung des Grundgesamtheitsmittels

$$SE_{\bar{X}} = \frac{s}{\sqrt{N}} \tag{16.10}$$

Mit diesem Wissen ausgestattet, können wir über die Schätzung des Standardfehlers der Objekthelligkeit nachdenken. Wir werden die folgenden drei Unsicherheitsquellen in Betracht ziehen: (1) zufälliges Rauschen innerhalb der Teleskopöffnung, das das thermische Rauschen des Sensors, das Ausleserauschen des Signalverstärkers und des Analog-Digital-Wandlers umfasst, (2) Poisson-Statistiken der Zählung diskreter Ereignisse (auf den Sensor auftreffende Photonen), die während eines festen Zeitraums auftreten, und (3) der Fehler bei der Schätzung der mittleren Himmelshelligkeit.

Für die Schätzung des mittleren Himmelsniveaus haben wir den Algorithmus des robusten Mittelwerts verwendet. Er ermöglicht die Schätzung seiner Stichprobenvarianz σ^2_{pxl}. Dies ist eine pixelbasierte Varianz, und da wir A Pixel in der Sternblende summiert haben, gilt die Bienaymé-Formel: Die Summe S ist eine Summe von A unkorrelierten Zufallsvariablen, von denen jede die Varianz σ^2_{pxl} hat. Für die Varianz der ersten Fehlerquelle erhalten wir:

$$\sigma_1^2 = A\,\sigma^2_{pxl} \tag{16.11}$$

wobei A eine Anzahl von Pixeln in der Sternblende ist.

Aus der Poisson-Statistik können wir eine Varianz ableiten, die durch die Zählung diskreter Ereignisse, der auf einen Sensor auftreffenden Photonen, entsteht, die während eines festen Zeitraums, der Belichtung, auftreten. Wir müssen wiederum die Verstärkung/Gain p des Sensors verwenden, um eine Intensität in ADU in eine Anzahl von Photonen umzuwandeln. Wenn die gemessene Nettointensität eines Objekts I ist, berechnen wir die mittlere Anzahl von Photonen $lambda$ als

$$\lambda = I\,p \tag{16.12}$$

Bemerkung

Der Wert der Verstärkung p des Sensors/ADC-Konverters kann im Abschnitt *Photometrie* der Siril-Einstellungen geändert werden

Dann ist die Varianz der Intensität aufgrund der Poisson-Statistik gleich ihrem Mittelwert.

$$\sigma^2_{ph} = \mathrm{Var}(\mathrm{Pois}(\lambda)) = \lambda = I\,p \tag{16.13}$$

Die Varianz ist in Photonen angegeben, wir müssen sie in ADU umrechnen, um die Varianz in Einheiten ADU^2 zu erhalten.

$$\sigma_2^2 = \frac{\sigma^2_{ph}}{p^2} = \frac{I\,p}{p^2} = \frac{I}{p} \tag{16.14}$$

Wir haben den Himmelspegel als Stichprobenmittelwert der Pixelpopulation im Himmelsring abgeleitet. Da jedes Pixel im Ring eine Varianz σ^2_{pxl} hat, ist die Varianz des Stichprobenmittelwertes

$$s^2_{sky} = \frac{\sigma^2_{pxl}}{n_{sky}} \tag{16.15}$$

wobei n_{sky} die Anzahl der Pixel im Himmelsring ist.

Aus der Gleichung (16.9) wird die Varianz der Intensität des Objekts berechnet als

$$\sigma^2_{ADU} = \sigma_1^2 + \sigma_2^2 + A^2\,s^2_{sky} \tag{16.16}$$

Beachten Sie, dass in der Gleichung (16.2) der Himmelspegel mit A multipliziert wird, so dass wir seine Varianz mit A^2 multiplizieren müssen - siehe die Gleichung (16.16). Nun verwenden wir das Gesetz der Fehlerfortpflanzung für den Logarithmus, der mit der Formel des Pogsonschen Gesetzes übereinstimmt.

$$\sigma_{mag}^2 = \left(\frac{-2.5}{I \ \log(10)} \right)^2 \sigma_{ADU}^2 \tag{16.17}$$

Setzt man die Gleichungen (16.17) und (16.16) zusammen, kann man den Standardfehler der Helligkeit des Objekts in Magnituden ableiten als

$$\sigma_{mag} = \frac{1.08574}{I} \sqrt{\sigma_{ADU}^2} \tag{16.18}$$

16.2 Schnelle Photometrie

16.2.1 Erklärung der Bereiche der Photometrie

Da Siril nur Aperturphotometrie durchführt, ist es wichtig, die richtigen Parameter zu kennen und einzustellen.

Die Sterne werden nach Ihrer Wahl modelliert: Gauß oder Moffat, wie im Kapitel *hier* im Kapitel *Dynamische PSF* beschrieben.

Ein bestimmter Stern auf einem bestimmten Bild hat also sein eigenes FWHM.

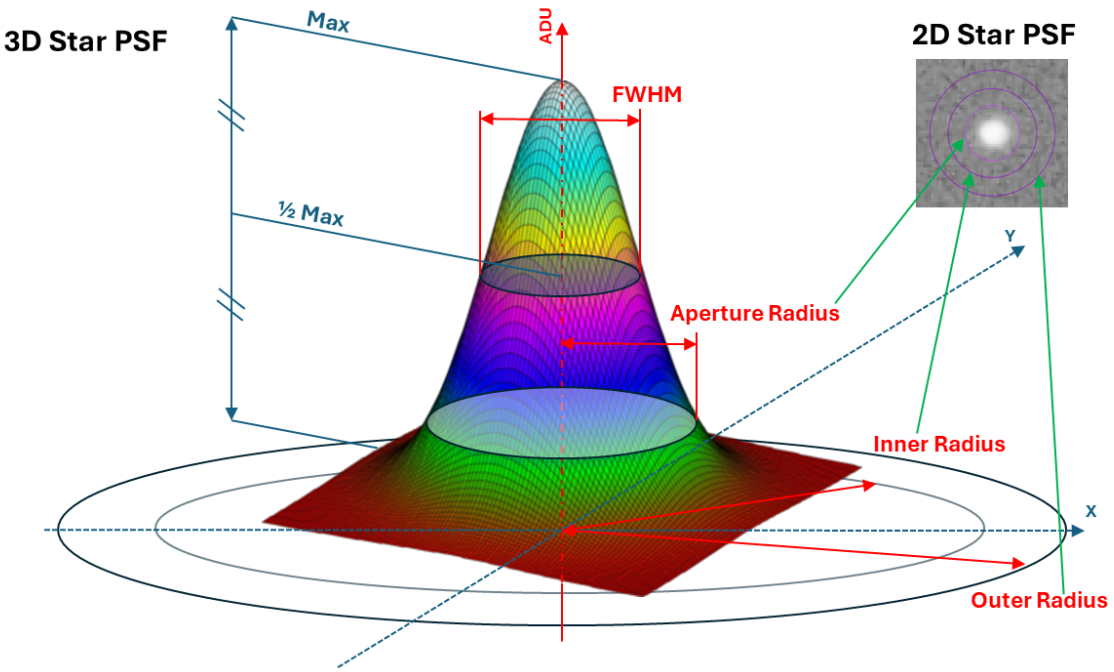


Abb. 3: PSF und Begriffe der Photometrie.

Für die Aperturphotometrie werden 3 Radien benötigt:

1. Der äußere Kreis, definiert durch den **Außenradius** (meist in Pixel ausgedrückt).
2. Der innere Kreis, definiert durch den **inneren Radius** (meist in Pixel ausgedrückt). Diese beiden Kreise definieren den Ring, der zur Messung des Hintergrunds verwendet wird (d. h. der Himmelspegel + zusätzliches Rauschen).

3. Der Aperturkreis, definiert durch den **Blendenradius**. Er kann in Pixel oder als Verhältnis zur FWHM des Sterns ausgedrückt werden (das automatische Kontrollkästchen im Fenster). Dieser Bereich wird verwendet, um das Sternsignal zu messen, das zu einem Hintergrundsignal hinzugefügt wird.

Diese 3 Radien können individuell über die GUI-Registerkarte *Einstellungen→Photometrie* eingestellt und optimiert werden.

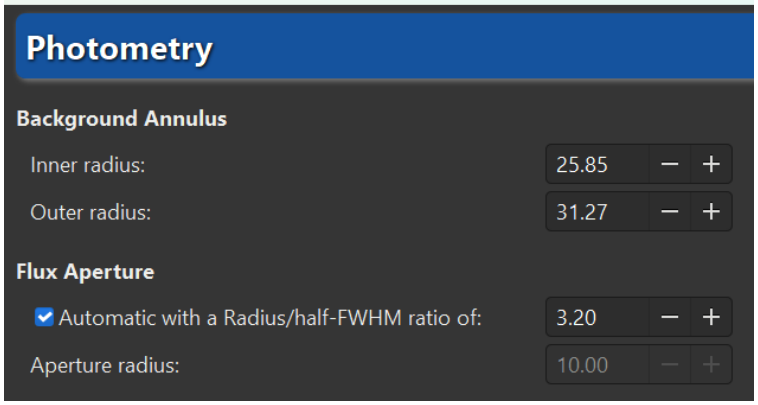


Abb. 4: Radiuseinstellungen.

Oder über die Befehlszeilenschnittstelle:

Siril Kommandozeile

```
setphot [-inner=20] [-outer=30] [-aperture=10] [-dyn_ratio=4.0] [-gain=2.3] [-min_
↪val=0] [-max_val=60000]
```

Liest oder setzt die Einstellungen für die Photometrie, die hauptsächlich von SEQPSF verwendet werden. Wenn Argumente angegeben werden, aktualisieren sie die Einstellungen. Keine der Einstellungen ist obligatorisch, es können alle optional angegeben werden, die Standardwerte sind in der Befehlssyntax angegeben. Am Ende des Befehls wird die aktive Konfiguration ausgedruckt.

Die Blendengröße ist dynamisch, es sei denn, sie wird erzwungen. Ist dies der Fall, wird der Wert **Blende** aus den Einstellungen verwendet. Wenn dynamisch, wird der Radius der Blende durch das angegebene dynamische Verhältnis („Radius/halber FWHM“) definiert.
Die zulässigen Werte für das Argument **-dyn_ratio** liegen im Bereich [1.0, 5.0]. Ein Wert außerhalb dieses Bereichs setzt die Blende automatisch auf den festen Wert **-aperture**.

Der Gainwert wird nur genutzt, wenn der nicht aus dem FITS-Header gelesen werden kann

Verweis: *seqpsf*

Tipp

Diese Radieneinstellungen gelten für alle photometrischen Verfahren mit Blende: **Schnellfotometrie** und **Lichtkurven**. Prüfen Sie sie sorgfältig.

16.2.2 Photometrie an handverlesenen Objekten eines einzelnen Bildes

Die Schaltfläche **Schnellphotometrie** ist eine Schaltfläche in der Symbolleiste, mit der Sie eine Photometrie der Sterne durchführen können, was im Allgemeinen der einfachste Weg ist.

Tipp

Wenn sich der Stern inmitten mehrerer Sterne befindet und das Werkzeug nicht auf den richtigen Stern zeigt, besteht eine alternative Möglichkeit darin, eine Auswahl um den Stern zu zeichnen und dann mit der rechten Maustaste auf *PSF* zu klicken. Es ist vielleicht auch interessant zu wissen, dass der mittlere Klick eine Auswahl mit einer empfohlenen Größe für PSF/Photometrie zeichnet (basierend auf dem eingestellten äußeren Radius).

Tipp

Wenn die Photometrie auf der RGB-Ebene durchgeführt wird, werden die Ergebnisse tatsächlich auf der grünen Ebene berechnet. Um Photometrie auf den roten oder blauen Schichten zu erhalten, müssen Sie an den entsprechenden Kanälen arbeiten.

Siril Kommandozeile

```
psf [channel]
```

Führt eine PSF (Point Spread Function) für den ausgewählten Stern durch und zeigt die Ergebnisse an. Im Headless-Betrieb kann die Auswahl mit BOXSELECT in Pixeln angegeben werden. Falls angegeben, wählt das Argument **channel** den Bildkanal aus, auf dem der Stern analysiert werden soll. Es kann bei monochromen Bildern oder bei der Ausführung über die grafische Benutzeroberfläche weggelassen werden, wenn einer der Kanäle in der Ansicht aktiv ist

Links: *boxselect*

Klicken Sie auf diese Schaltfläche, um den Bildauswahlmodus zu ändern, und klicken Sie dann auf einen Stern. Es wird nicht nur die Photometrie berechnet, sondern Siril berechnet auch die PSF (Point Spread Function/Punktspreizfunktion) des Sterns, mit einer Vielzahl von Details.

Für die Berechnung der PSF werden zwei Berechnungsmodelle verwendet, die vom Benutzer im Fenster *Dynamisches PSF* ausgewählt werden können (`Strg` + `F6`).

Das Ergebnis der Photometrie und die zugehörige PSF werden in folgender Form angezeigt:

```
PSF fit Result (Gaussian, monochrome channel):

Centroid Coordinates:
            x0=5258.25px     09h25m34s J2000
            y0=2179.72px    +69°49'31" J2000

Full Width Half Maximum:
            FWHMx=7.13"
            FWHMy=6.79"
```

(Fortsetzung auf der nächsten Seite)

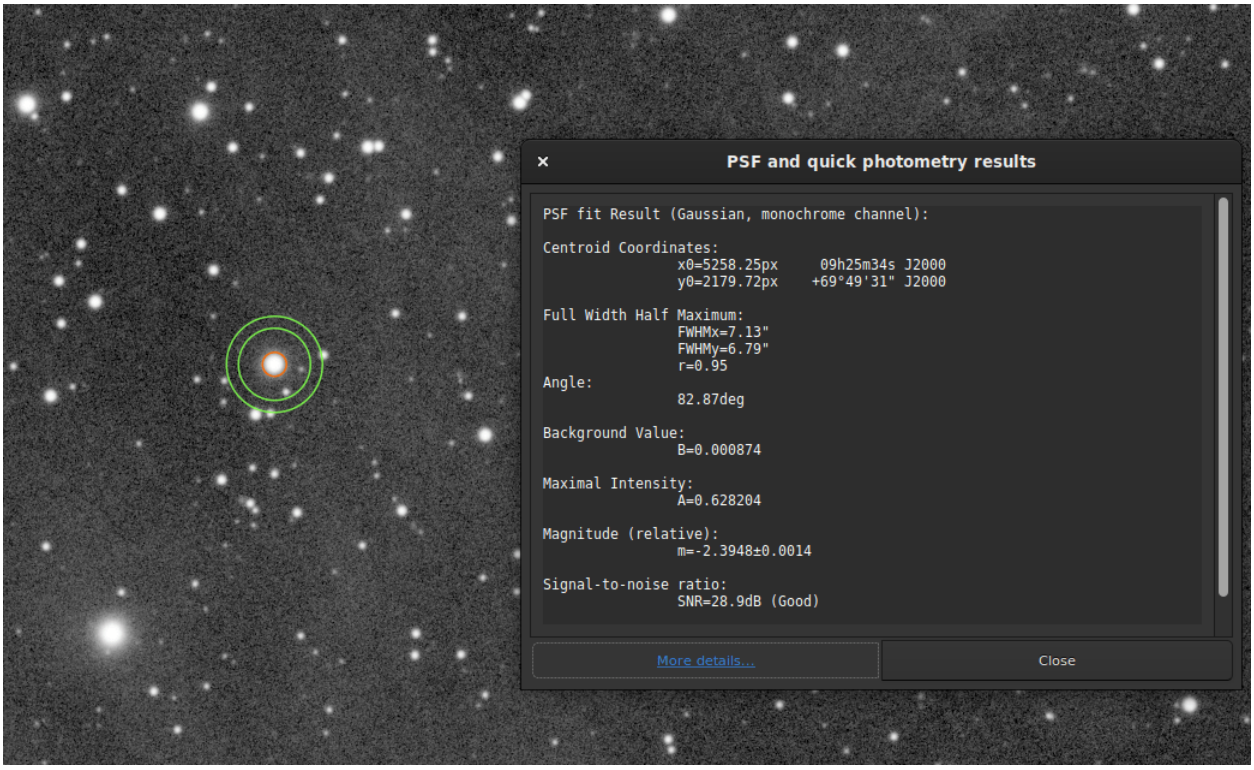


Abb. 5: Fenster mit photometrischem Ergebnis.

(Fortsetzung der vorherigen Seite)

```
            r=0.95
Angle:
            82.87deg

Background Value:
            B=0.000874

Maximal Intensity:
            A=0.628204

Magnitude (relative):
            m=-2.3948±0.0014

Signal-to-noise ratio:
            SNR=28.9dB (Good)

RMSE:
            RMSE=1.890e-03
```

1. Die Angleichung wurde mit der **Gaußschen** Anpassungsfunktion durchgeführt, so dass keine zusätzlichen Parameter erforderlich sind. Wenn jedoch Moffat verwendet wurde, wird die folgende Ausgabe angezeigt:

   ```
   PSF fit Result (Moffat, beta=2.9, monochrome channel):
   ```

2. **Schwerpunktkoordinaten** (Centroid Coordinates) gibt die Koordinaten des Schwerpunkts in Pixeln an. Wenn

jedoch, wie im obigen Beispiel, eine astrometrische Lösung das Bild existiert, gibt Siril die Koordinaten im Welt-Koordinatensystem (RA und Dek) an.

3. **Full Width Half Maximum** (FWHM/Halbwertsbreite) wird in Bogensekunden ausgegeben, wenn der Bildmaßstab bekannt ist (aus dem Header oder aus der *Bildinformationen* → *Information*) und in Pixeln, falls nicht. Die Rundheit r wird als das Verhältnis von $\frac{\text{FWHMy}}{\text{FWHMx}}$ errechnet.
4. **angle** (Winkel) ist der Drehwinkel der X-Achse in Bezug auf die Schwerpunktkoordinaten. Er variiert in dem Bereich $[-90, +90]$.
5. **Background Value** (Hintergrundwert) ist der lokale Hintergrund im Bereich $[0, 1]$ für 32-Bit-Bilder und $[0, 65535]$ für 16-Bit-Bilder. Es handelt sich dabei um einen angepassten Wert, nicht um den im Ring der Blendenphotometrie berechneten Hintergrund.
6. Der Wert **maximum Intensity** (Maximalintensität) ist ebenfalls ein angepasster Wert und stellt die Amplitude dar. Es handelt sich um den Maximalwert der angepassten Funktion, der sich an den Koordinaten des Schwerpunkts befindet.
7. Die **Magnitude**, die mit ihrer Genauigkeit angegeben wird, ist das Ergebnis der Photometrie. Wenn die Berechnung jedoch aus bestimmten Gründen nicht möglich ist (gesättigte Pixel oder schwarze Pixel), wird eine Unsicherheit von **9,999** angegeben. In diesem Fall wird die Photometrie als ungültig gekennzeichnet, aber es wird immer noch ein Magnitudenwert angegeben, der jedoch mit Vorsicht verwendet werden sollte.
8. In den Ergebnissen wird ein Schätzwert für das **signal-to-noise ratio** (Signal-Rausch-Verhältnis) angezeigt. Sein Wert wird nach der folgenden Formel berechnet und in dB angegeben:

$$\text{SNR} = 10 \log_{10} \left(\frac{I}{N} \right) \tag{16.19}$$

   wobei I die Nettointensität ist, proportional zum beobachteten Fluss F und N die Summe der Unsicherheiten, wie in (16.18) ausgedrückt.

   Zum besseren Verständnis wird sie in 6 Qualitätsstufen eingeteilt:

   1. Hervorragend (SNR > 40dB)
   2. Gut (SNR > 25dB)
   3. Befriedigend (SNR > 15dB)
   4. Niedrig (SNR > 10dB)
   5. Schlecht (SNR > 0dB)
   6. Unbekannt

      Dieser letzte Vermerk wird nur angezeigt, wenn die Berechnung aus dem einen oder anderen Grund fehlgeschlagen ist.
9. Schließlich liefert **RMSE** einen Schätzwert für die Anpassungsqualität. Je niedriger der Wert, desto besser das Ergebnis.

Wenn eine photometrische Lösung für das Bild gefunden wird, führt die Schaltfläche *Mehr Details* am unteren Rand des Fensters zu einer Seite auf der SIMBAD-Website mit Informationen über den ausgewählten Stern. Es ist jedoch möglich, dass die Seite keine zusätzlichen Informationen liefert, wenn der Stern nicht in der SIMBAD-Datenbank enthalten ist.

16.2.3 Schnelle Photometrie von Sequenzen

Eine Schnellphotometrie kann auch mit einer Sequenz durchgeführt werden. Dies ist im Allgemeinen dazu gedacht, eine Lichtkurve zu erhalten, wie *hier* erklärt. Um dies durchzuführen, müssen Sie **eine Sequenz laden**, eine Auswahl um einen Stern zeichnen und dann **mit der rechten Maustaste** auf das Bild klicken.

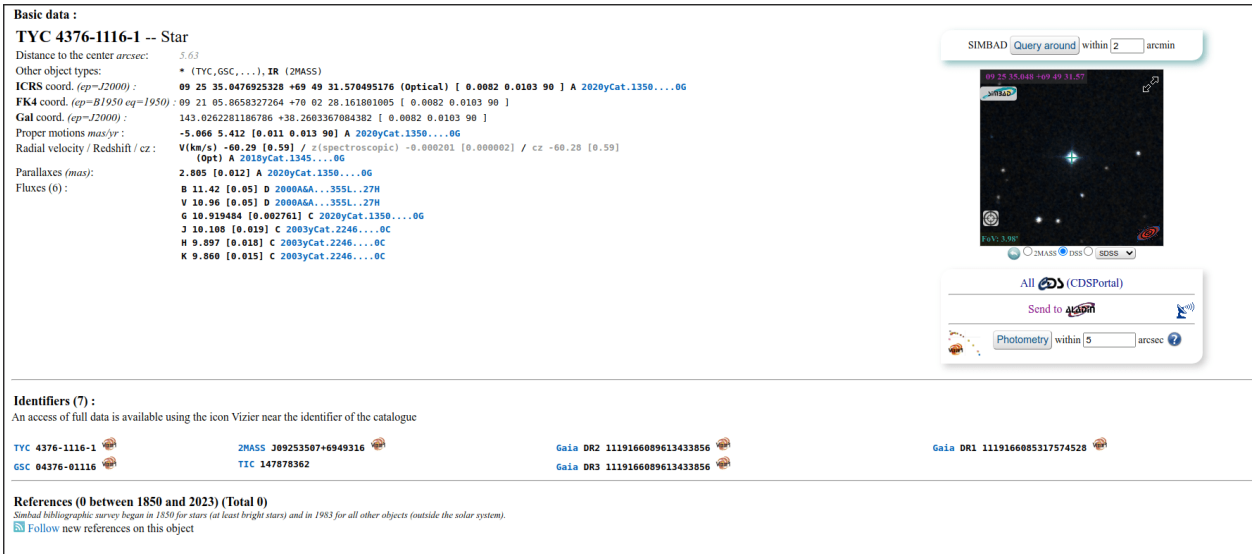


Abb. 6: Mehr Details über den analysierten Stern. Klicken Sie auf das Bild, um es zu vergrößern.

> **Tipp**
>
> Im Idealfall muss die Sequenz ohne Interpolation registriert werden, um die Rohdaten nicht zu verändern. Verwenden Sie z. B. die **globale Sternausrichtung** mit der Option **Nur Transformation in der Sequenzdatei speichern**.

> **Bemerkung**
>
> Stellen Sie sicher, dass die inneren und äußeren Radien für den Hintergrundring an den zu analysierenden Stern und die Sequenz angepasst sind. Einige Bilder können aufgrund von Himmelsbedingungen oder schlechter Nachführung eine viel größere FWHM haben als das Referenzbild. Sie können in den *Einstellungen* oder mit dem Befehl `setphot` geändert werden.

Am Ende des Prozesses öffnet Siril automatisch die Registerkarte "Grafischer Plot" mit den berechneten Kurven. Es ist möglich, auf mehrere Sterne zu klicken, um die Berechnung zu reproduzieren, wobei der erste Stern den besonderen Status eines *veränderlichen Sterns* erhält und die übrigen Sterne als *Referenzen* dienen. Dies ist wichtig für die Berechnung der Lichtkurve.

16.2.4 Berechnung der wahren Größenordnungen

Die berechnete Helligkeit ist nur aussagekräftig, wenn sie mit anderen Werten im linearen Bild verglichen wird. Der angegebene Wert entspricht nicht der tatsächlichen sichtbaren Helligkeit des Sterns, er ist unkalibriert, auch relative Helligkeit genannt.

Siril stellt Werkzeuge zur Verfügung, mit denen eine ungefähre scheinbare Helligkeit berechnet werden kann. Dazu muss die Helligkeit eines anderen auf dem Bild sichtbaren Sterns bekannt sein, der als Referenz dienen soll. Derzeit ist es nur möglich, einen einzigen Stern als Referenz zu verwenden, daher die *ungefähre* Angabe. Für eine höhere Genauigkeit sollte ein Stern mit ähnlicher Farbe und Helligkeit wie der/die zu messende(n) Stern(e) gewählt werden, und seine Helligkeit sollte mit dem für die Aufnahme des Bildes verwendeten Filter übereinstimmen. Kataloge enthalten Helligkeiten, die mit einem <https://en.wikipedia.org/wiki/Photometric_system>`_ photometrischen Filter berechnet wurden, der im Allgemeinen nicht von Amateuren verwendet wird, um schöne Bilder zu machen, was eine weitere Annäherung darstellt.

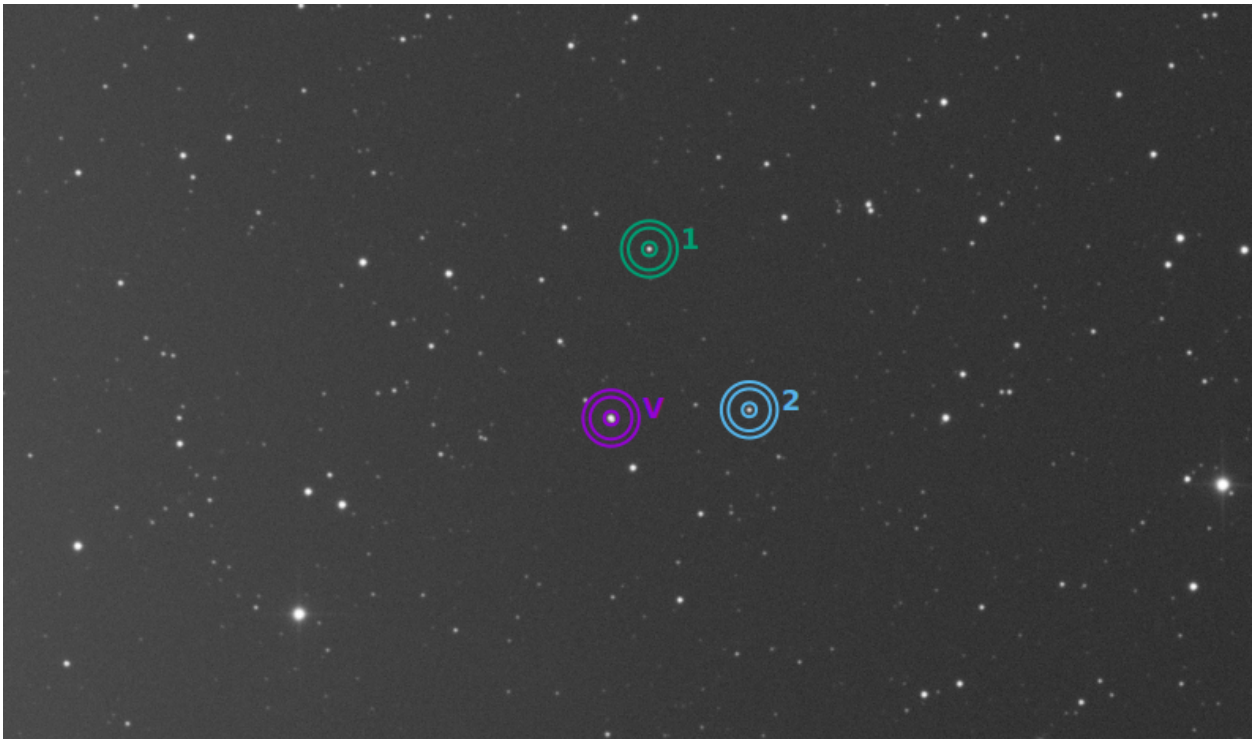


Abb. 7: In diesem Beispiel wurden 3 Sterne analysiert. Der erste wird als veränderlicher Stern verwendet. Die anderen sind Referenzen.

- Führen Sie eine schnelle Photometrie an einem bekannten Stern durch, die angegebene relative Helligkeit ist `-2.428`. Es ist möglich, die tatsächliche sichtbare Helligkeit herauszufinden, indem Sie auf die Schaltfläche *Mehr Details* klicken, wie oben beschrieben. Sagen wir, der gefundene Wert ist `11.68` (stellen Sie sicher, dass Sie einen Wert verwenden, der dem Spektralband des Bildes entspricht).
- Lassen Sie dann den Stern markiert und geben Sie in Siril folgenden Befehl ein

```
setmag 11.68
```

Das ergibt dann etwa folgende Ausgabe

```
10:50:49: Relative magnitude: -2.428, True reduced magnitude: 11.680,␣
↪Offset: 14.108
```

Siril Kommandozeile

```
setmag magnitude
```

Kalibriert die Helligkeit durch Auswahl eines Sterns und Angabe der bekannten scheinbaren Magnitude.

Alle PSF-Berechnungen liefern anschließend die kalibrierte scheinbare Magnitude anstelle einer scheinbaren Helligkeit relativ zu ADU-Werten. Beachten Sie, dass der angegebene Wert mit der Helligkeit übereinstimmen muss, damit der Beobachtungsfilter sinnvoll ist.

Zum Zurücksetzen der Magnitudenkonstante siehe UNSETMAG

Verweis: *psf*, *unsetmag*

- Nun müssen alle berechneten Magnituden Werte aufweisen, die nahe an ihrer tatsächlichen visuellen Helligkeit liegen. Dies gilt jedoch insbesondere für Sterne, deren Helligkeit in der gleichen Größenordnung liegt wie die des Referenzsterns.

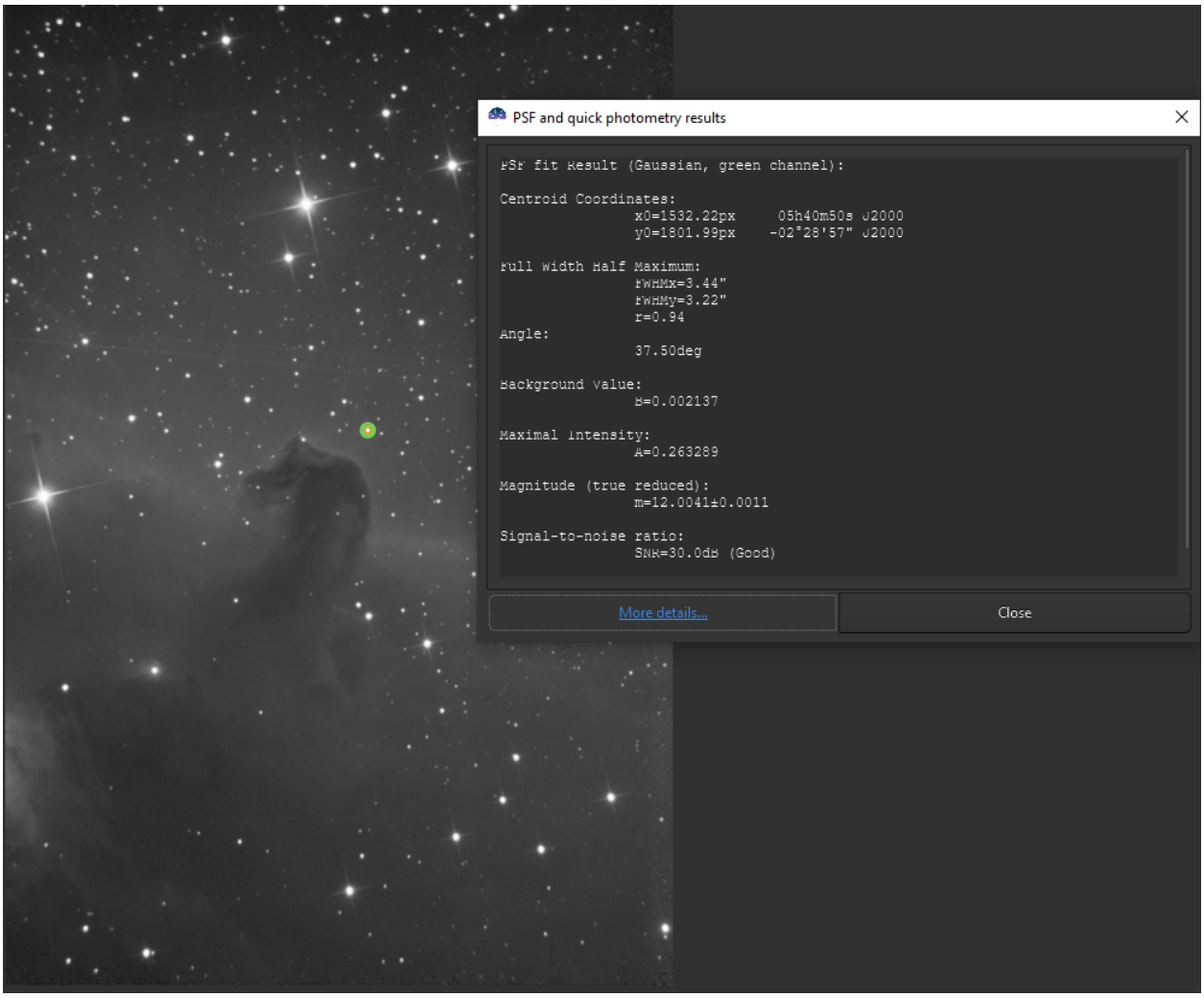


Abb. 8: Fenster mit den Ergebnissen der Photometrie mit eingestellter wahrer Magnitude.

- Um den berechneten Offset wieder zu löschen, geben Sie einfach

```
unsetmag
```

Siril Kommandozeile

```
unsetmag
```

Setze die Magnituden-Kalibrierung auf 0 zurück. Vgl. SETMAG

Verweis: *setmag*

Tipp

Die gleichen Befehle gibt es auch für die Sequenzen. Sie lauten `seqmagset` und `sequnsetmag`. Sie werden auf die gleiche Weise verwendet, wenn eine Sequenz geladen wird.

Siril Kommandozeile

```
seqsetmag magnitude
```

Wie der Befehl SETMAG, aber für die geladene Sequenz.

Dieser Befehl ist nur gültig, nachdem Sie SEQPSF oder sein grafisches Gegenstück ausgeführt haben (wählen Sie den Bereich um einen Stern aus und starten Sie die PSF-Analyse für die Sequenz; sie wird in der Registerkarte "Grafischer Plot" angezeigt).
Dieser Befehl hat das gleiche Ziel wie SETMAG, berechnet aber die Referenzgröße für jedes Bild der Sequenz, in der der Referenzstern gefunden wurde, neu.
Beim Ausführen des Befehls wird der zuletzt analysierte Stern als Referenzstern betrachtet. Die Anzeige des Magnituden-Diagramms vor der Eingabe des Befehls erleichtert das Verständnis.
Um den Referenzstern und den Magnituden-Offset zurückzusetzen, siehe SEQUNSETMAG

Verweise: *setmag*, *seqpsf*, *psf*, *sequnsetmag*

Siril Kommandozeile

```
sequnsetmag
```

Setzt die Magnitudenkalibrierung und den Referenzstern für die Sequenz zurück. Siehe SEQSETMAG

Verweis: *seqsetmag*

16.3 Lichtkurven

In der Astronomie ist eine Lichtkurve ein Diagramm der Lichtintensität eines Himmelsobjekts in Abhängigkeit von der Zeit, typischerweise mit der Magnitude des empfangenen Lichts auf der y-Achse und der Zeit auf der x-Achse. Siril ist in der Lage, solche Kurven bei der Analyse von Sternen zu erstellen.

Es gibt nun zwei Möglichkeiten, die veränderlichen Sterne und die Referenzsterne (auch Vergleichssterne genannt) auszuwählen: manuell oder mit Hilfe einer Liste von Sternen, die vom N.I.N.A.-Exoplaneten-Plugin erhalten wurde.

16.3.1 Manuelle Sternwahl

Wechseln Sie zunächst in den **Photometrie-Modus** (klicken Sie auf).

Wählen Sie (Rechtsklick) den veränderlichen Stern aus, gefolgt von den gewünschten Vergleichssternen, wie *hier* erklärt.

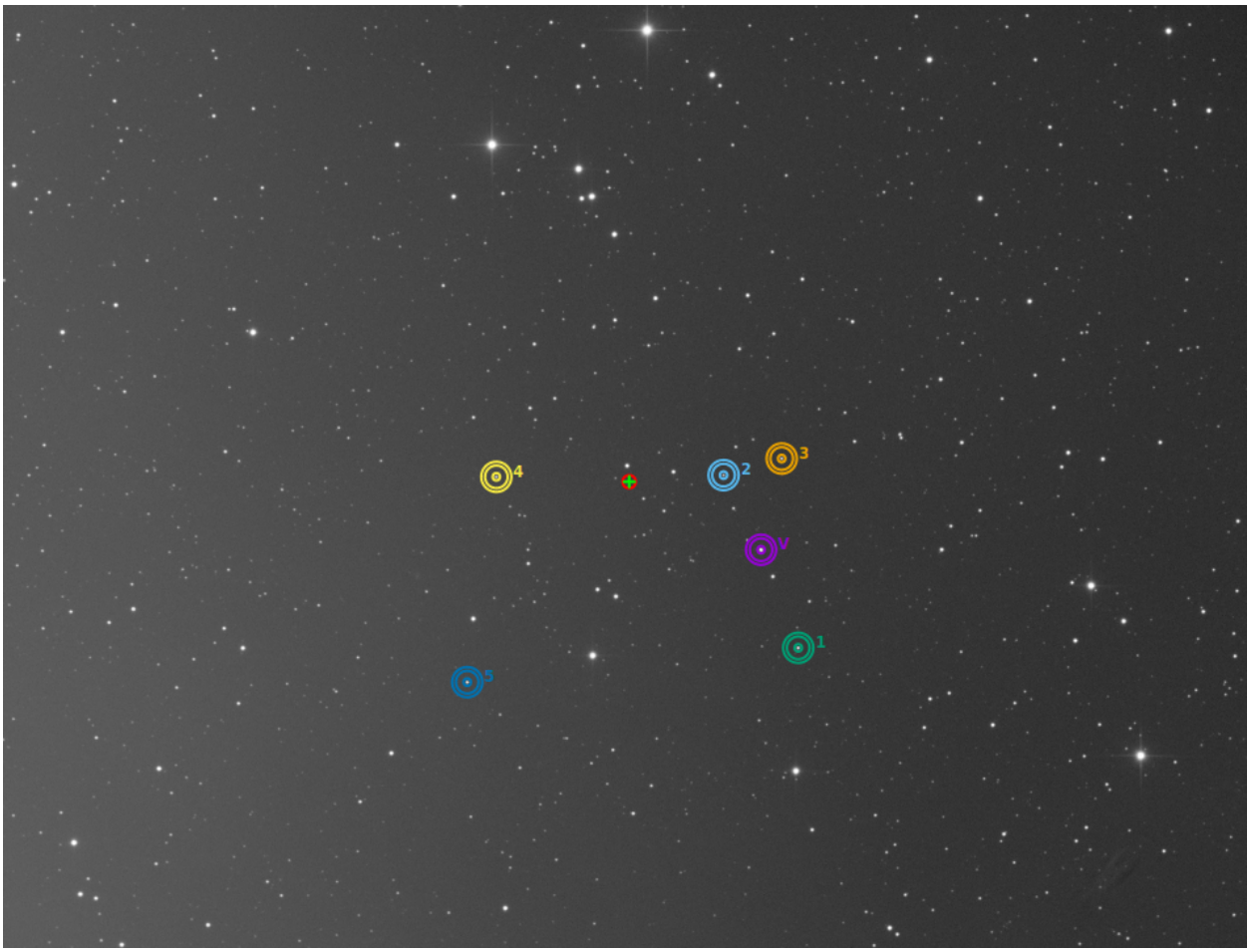


Abb. 9: Ein Stern ist der Veränderliche (violett mit einem V) und die 5 anderen dienen als Referenz.

Warnung

Achten Sie darauf, dass Sie keine veränderlichen Sterne als Referenz auswählen. Wenn die Astrometrie auf Ihrem Bild durchgeführt wird, zögern Sie nicht, die *SIMBAD Anfrage* zu verwenden, um mehr über die Sterne zu erfahren.

Eine praktische Möglichkeit, Ihren Vergleichsstern zu überprüfen, besteht darin, die drei Variablenkataloge für veränderliche Sterne anzuzeigen: GCVS, AAVSO-VSX und GAIA-varisum.

Sie müssen Ihren eigenen Parameter (Grenzgröße) festlegen, aber als Beispiel sind hier die Befehle:

```
conesearch 18.0 -cat=gcvs
conesearch 18.0 -cat=aavso_chart
conesearch 18.0 -cat=varisum
```

> **Tipp**
>
> Es ist vorzuziehen, Referenzen zu wählen, deren Magnitude nahe an der des Variablen Sterns liegt, wie später *hier* erläutert.

Sobald dies geschehen ist, lädt Siril automatisch die Registerkarte "Grafischer Plot", wie in der Abbildung unten dargestellt. Hier werden die FWHM-Kurven als Funktion der Bildnummer dargestellt.

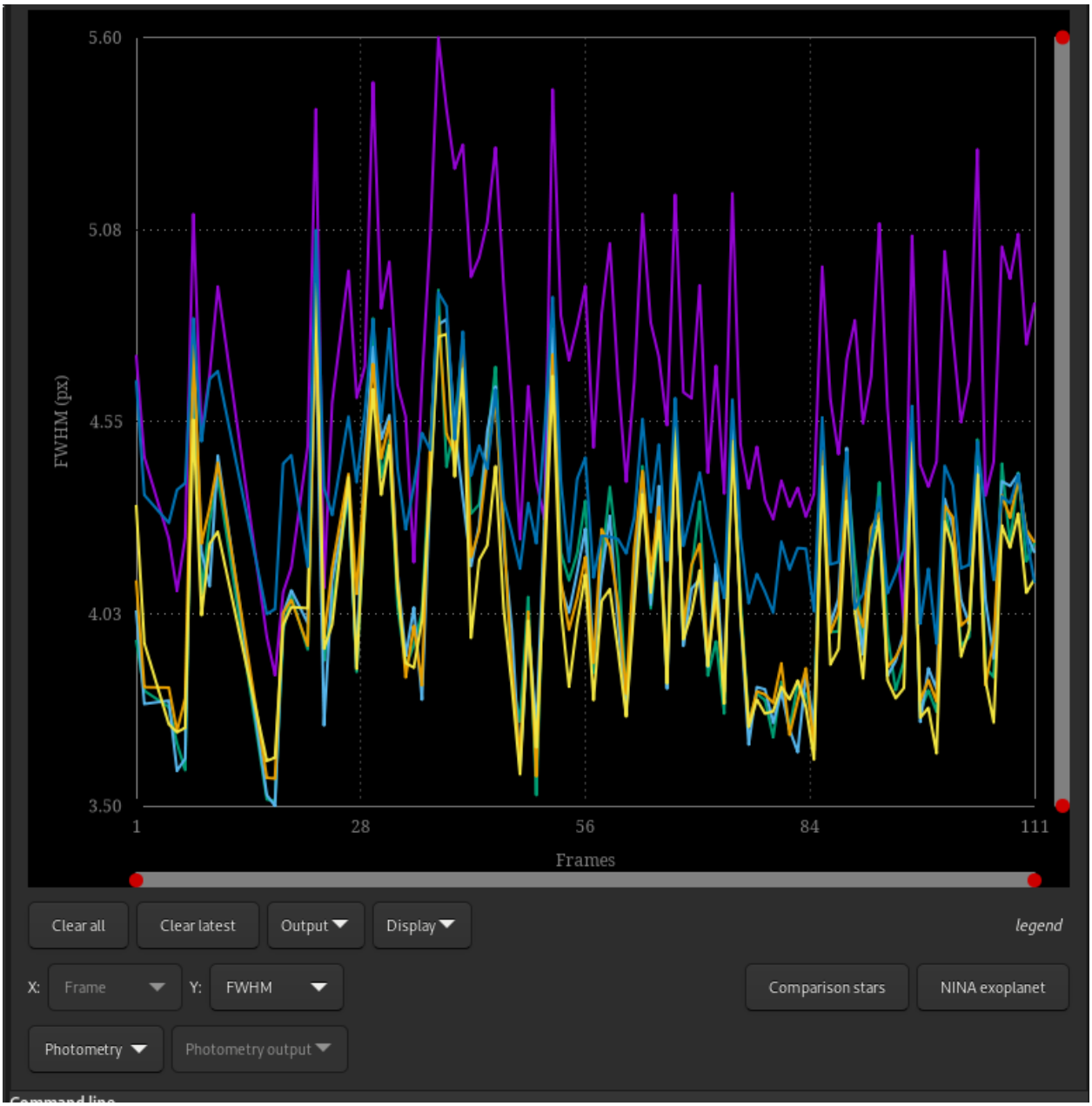


Abb. 10: Die Registerkarte "Grafischer Plot" wird direkt nach der Schnellphotometrie der Sequenz angezeigt.

Was uns in diesem Teil interessiert, ist die Darstellung der Magnitudenkurven. Gehen Sie einfach zum Dropdown-Menü und ändern Sie **FHWM** in **Magnitude**. Anschließend werden die Magnitudenkurven jedes analysierten Sterns angezeigt. Die Untermenüs in *Photometrie* des *Werkzeug*-Menüs sind ebenfalls aktiv.

Ausgewählte Vergleichssterne speichern

Wenn Sie mit den eingestellten Vergleichssternen zufrieden sind, können Sie diese für die zukünftige Verwendung in einem *Automatische Lichtkurve*-Prozess speichern oder einfach eine Liste zur weiteren Dokumentation aufbewahren.

Wählen Sie im *Extras*-Menü :menuselection:`Photometrie --> Vergleichssterndatei erstellen...“.

Die Standardeinstellung *Sterne im aktuell geladenen Bild verwenden* ist ausgewählt.

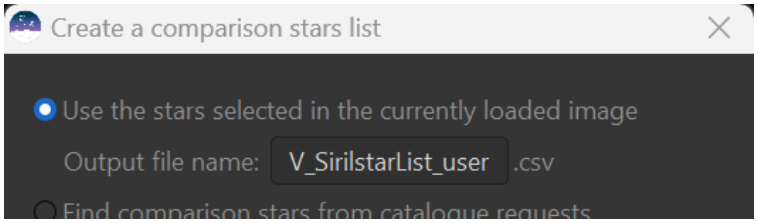


Abb. 11: Die manuell ausgewählten Sterne in einer .csv-Datei speichern.

Der Standardname der Ausgabedatei lautet `V_SirilstarList_user.csv`, Sie können ihn jedoch nach Bedarf ändern.

Nachdem Sie auf *OK* geklickt haben, wird die Datei erstellt und in Ihrem aktuellen Arbeitsverzeichnis gespeichert.

Lichtkurve (ETD-Format)

Die Schaltfläche *Lichtkurve (ETD-Format)* im Menü *Extras* → *Photometrie*→ *Ausgabe erstellen* ist für Exoplanetentransitdaten bestimmt und erzeugt eine Datei in dem von der ETD-Website <http://var2.astro.cz/ETD>`_ verwalteten Format.

Sobald die Analyse mit einer Anzahl mindestens 4 oder 5 Referenzsternen abgeschlossen ist (je höher die Zahl, desto genauer das Ergebnis. Es sind jedoch maximal 19 Sterne zulässig). Siril fragt nach einem Dateinamen, um die Daten im „csv“-Format zu speichern, dann wird die Lichtkurve in einem neuen Fenster angezeigt.

```
#JD_UT (+ 2457403)
# JD_UT V-C err
2457403.260428 -1.01054 0.00468598
2457403.261956 -1.00321 0.0046727
2457403.263461 -1.00132 0.00540513
2457403.264977 -0.994067 0.00505492
2457403.266481 -1.00397 0.00465623
2457403.267998 -1.00532 0.00467666
...
```

Dies ist eine Datei mit 3-Spalten. Die ersten beiden Zeilen beginnen mit einem „#“ und sind Kommentare. Dann gibt die erste Spalte den genauen Zeitpunkt der Beobachtung im Julianischen Datumsformat an. Die zweite Spalte ist die Helligkeit des Variablen (V), von der die Summe der instrumentellen Helligkeiten der Vergleichssterne abgezogen wurde. Die dritte Spalte stellt die Größenunsicherheit dar.

Theorie

V wird durch die folgende Formel berechnet:

$$V = (V_{\text{ins}} - C_{\text{ins}}), \tag{16.20}$$

mit

$$C_{\text{ins}} = -2.5 \log_{10} \left(\frac{1}{N} \sum 10^{-0.4 C_{\text{ins},i}} \right). \tag{16.21}$$

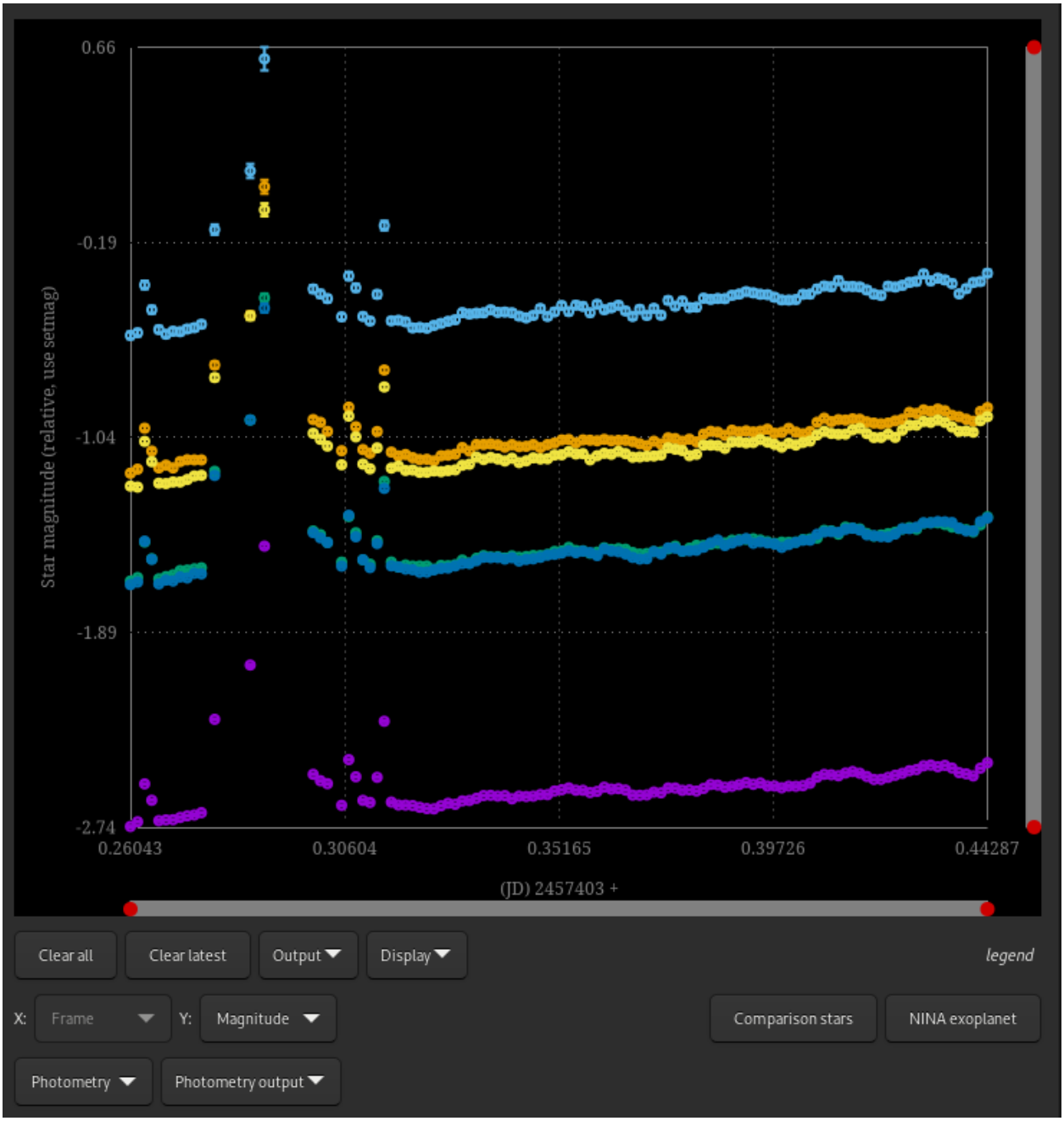


Abb. 12: Beim Umschalten auf die Magnitudenansicht wird die Schaltfläche *Photometrische Ausgabe* aktiv.

N ist die Gesamtzahl der Vergleichssterne und C_{ins}, i ist die instrumentelle Größe des i-ten Sterns im Ensemble.

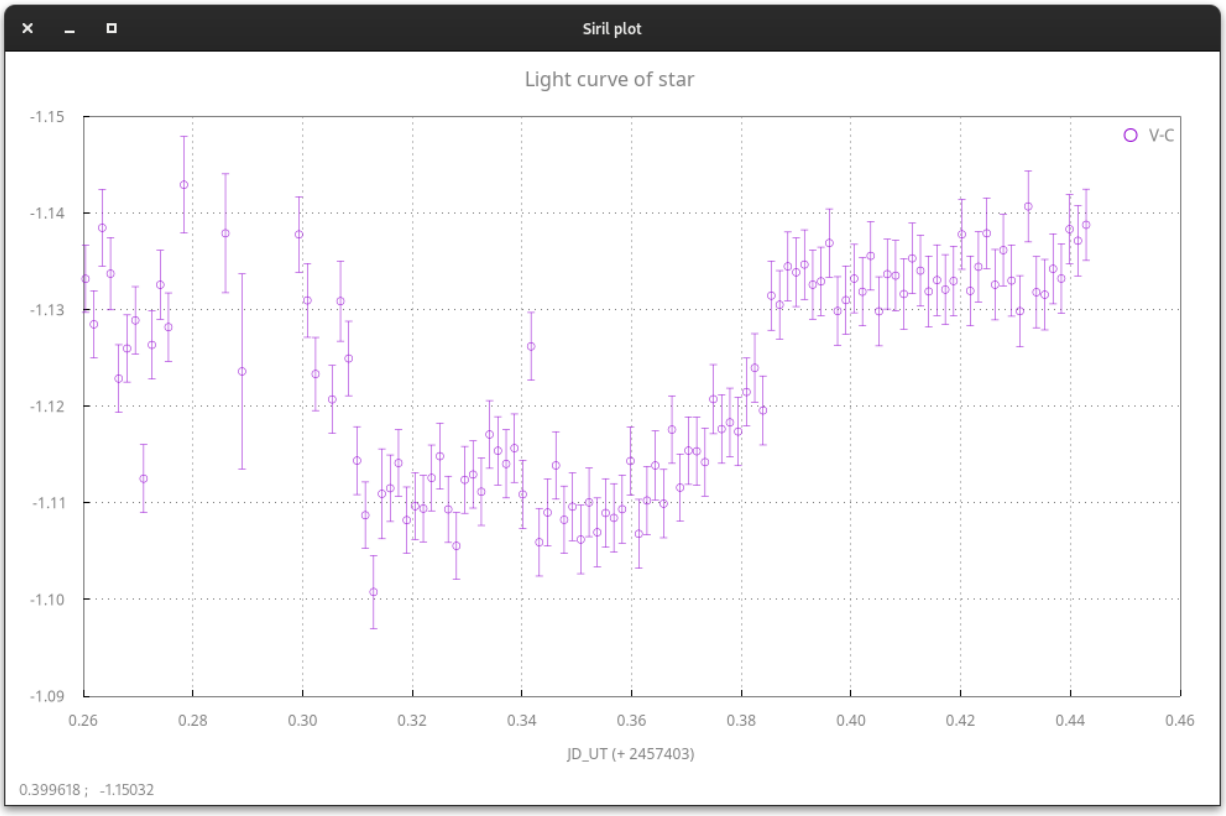


Abb. 13: Lichtkurve eines Exoplaneten-Transits.

Tipp

Die Lichtkurve wird mit dem internen Plottool von Siril gezeichnet. Ausführlichere Anweisungen zur Verwendung finden Sie in der *Dokumentation*.

Erweitertes AAVSO-Dateiformat

Die Schaltfläche *AAVSO-Dateiformat* befindet sich im Menü *Extras* → *Photometrie* → *Ausgabe erstellen* und ist dem erweiterten AAVSO-Dateiformat für veränderliche Sterne gewidmet. Dadurch wird ein neuer Dialog geöffnet, der nach Abschluss eine CSV-Datei exportiert, die auf die entsprechende Website hochgeladen werden kann.

Theorie

In Siril verwendet das exportierte AAVSO-Dateiformat die standardisierte Magnitude, die wie folgt definiert ist:

$$V_{\text{std}} = (V_{\text{ins}} - C_{\text{ins}}) + C_{\text{std}} \tag{16.22}$$

wobei V_{ins} und C_{ins} die instrumentelle Magnitude des Veränderlichen bzw. der Vergleichssterne, und C_{std} die Diagramm-Magnitude für den Vergleich sind. Sie können den Wert für C_{std} im AAVSO-Parameter-Dialog vor dem Export angeben.

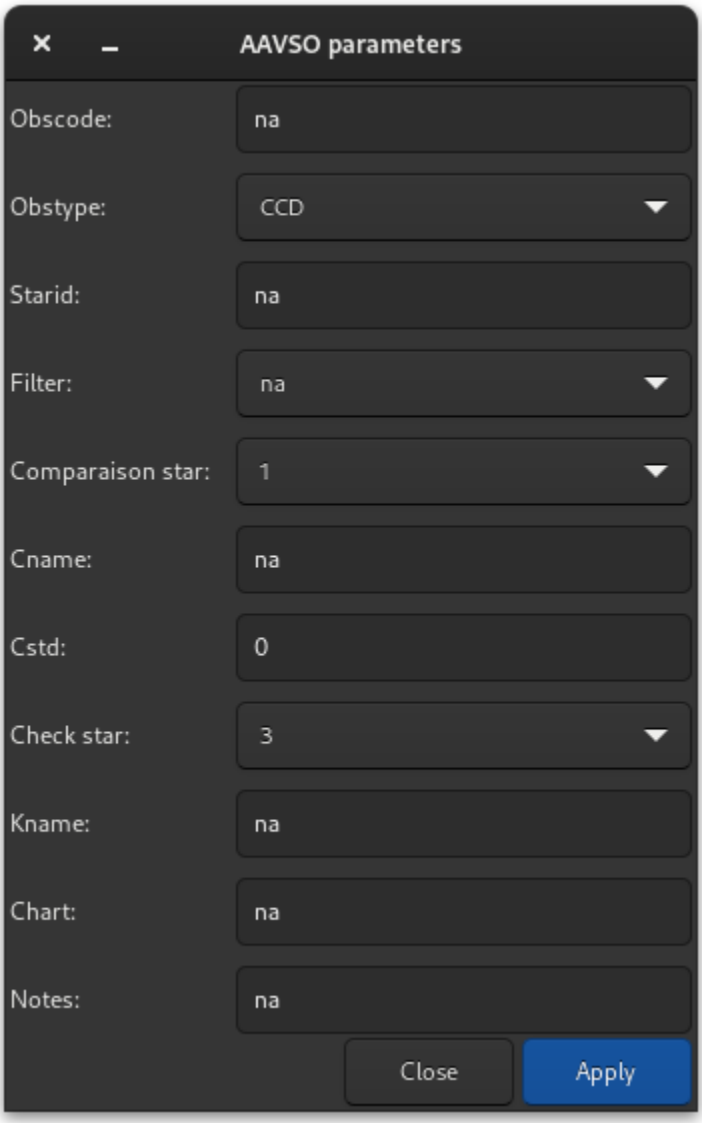


Abb. 14: AAVSO-Parameter-Sialog. Dieser Dialog wird zum Füllen der erweiterten AAVSO-Datei verwendet. Weitere Informationen finden Sie auf der „AAVSO-Website <https://www.aavso.org/aavso-extended-file-format>“

Ein Beispiel für die Photometrie ist unten mit dem Stern Rx And angegeben. Dieser Stern weist eine große Variabilität auf: Seine Helligkeit variiert zwischen 10,2 und 15,1 mit einer Periodizität von etwa 13 Tagen. Nach der Verarbeitung in Siril, Festlegung eines Vergleichssterns und eines Prüfsterns erhalten wir nach dem Hochladen der Daten die folgende AAVSO-Kurve. Die neuen Daten befinden sich ganz rechts (aufgenommen am 24. und 25. November 2023) und werden grün dargestellt.

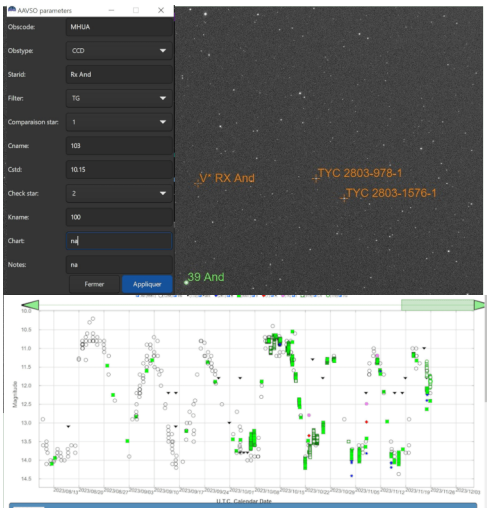


Abb. 15: Ein konkretes Beispiel für die Photometrie variabler Sterne (mit freundlicher Genehmigung von H. Meunier).

16.3.2 Automatische Sternauswahl

Automatische Lichtkurve

Um den Prozess der Transitanalyse von Exoplaneten zu automatisieren, können Listen von Referenzsternen, auch Vergleichssterne genannt, aus Sternkatalogen entnommen werden, mit den entsprechenden Kriterien: ähnliche Helligkeit/Magnitude, ähnliche Farbe (damit sich ihre relative Helligkeit nicht durch die atmosphärische Extinktion in verschiedenen Höhenlagen ändert), räumliche Nähe.

Die Aufnahmesoftware N.I.N.A verfügt über ein Exoplaneten-Plugin, das solche Sterne anzeigt und die Speicherung der Liste in einer CSV-Datei ermöglicht, z. B. `csv-Datei`:

```
Type,Name,HFR,xPos,yPos,AvgBright,MaxBright,Background,Ra,Dec
Target,HD 189733 b,2.6035068712769851,1992,1446,1640.3703703703704,39440,1917.
↪0601851851852,300.18333333333328,22.709722222222222
Var,SW Vul,2.8626145609282911,2972,276,26.14,2012,1905.445,300.02171,22.93517
Var,DQ Vul,2.372369130017419,3006,1040,28.180555555555557,2048,1906.9027777777778,300.
↪01254,22.78103
Var,HQ Vul,3.8351043206620834,157,1690,49.393939393939391,2104,1905.7454545454545,300.
↪55808,22.64067
...
Comp1,ATO J300.3222+22.7056,2.4268101078425852,1367,1465,352,4496,1913.9504132231405,300.
↪32229415181337,22.705681453738887
Comp1,HD 189657,2.5343988482845927,2527,2808,23.814814814814813,2012,1906.5061728395062,
↪300.08714683055996,22.4400393728
...
Comp2,000-BJP-946,2.2738807043120195,1832,750,29.962962962962962,2024,1910.0648148148148,
↪300.23741666666666,22.846999999999998
Comp2,000-BJP-942,2.0977710589704297,2760,1572,31.083333333333332,2096,1908.
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
↪6527777777778,300.025875,22.704777777777778
...
```

Im *Werkzeug*-Menü kann Siril diese Datei über die Schaltfläche *Automatische Lichtkurve...* laden. Um dies nutzen zu können, müssen einige Voraussetzungen erfüllt sein:

- die Sequenz der kalibrierten Bilder muss bereits geladen sein
- das Referenzbild der Sequenz muss astrometrisch gelöst werden, um sicherzustellen, dass wir die richtigen Sterne anhand ihrer äquatorialen J2000-Koordinaten identifizieren

Von da an läuft alles automatisch ab und zeigt am Ende des Prozesses die Lichtkurve für die ausgewählten Vergleichssterne.

In diesem Fenster werden auch die verwendeten *Blenden- und Hintergrundbereiche* angezeigt.

Das folgende Video zeigt eine automatische Verarbeitung der Lichtkurve mit der Vergleichssternliste von NINA:

Lichtkurve durch Befehl

Es ist auch möglich, die Lichtkurve mit dem Befehl `light_curve` zu automatisieren oder remote zu erstellen. Da der Blindbetrieb so weit wie möglich automatisiert werden muss, kann die Konfiguration der Radien des Hintergrundringes mit dem Argument `-autoring` automatisiert werden: Es wird eine Sternsuche im Referenzbild durchgeführt und die mittlere FWHM mit einem konfigurierbaren Faktor multipliziert, um die inneren und äußeren Radien zu erhalten, die mit der Sequenz funktionieren sollten.

Siril Kommandozeile

```
light_curve sequencename channel [-autoring] { -at=x,y | -wcs=ra,dec } { -refat=x,y |␣
↪-refwcs=ra,dec } ...
light_curve sequencename channel [-autoring] -ninastars=file
```

Analysiert mehrere Sterne mit der Aperturphotometrie in einer Bildsequenz und erstellt eine Lichtkurve für einen Stern, die durch die anderen kalibriert wird. Die ersten Koordinaten, in Pixeln, wenn **-at=** verwendet wird, oder in Grad, wenn **-wcs=** verwendet wird, sind für den Stern, dessen Licht aufgezeichnet werden soll, die anderen für die Vergleichssterne.

Alternativ kann eine Liste von Ziel- und Referenzsternen im Format der Sternliste des NINA Exoplaneten-Plugins mit der Option **-ninastars=** übergeben werden. Siril prüft, ob alle Referenzsterne verwendet werden können, bevor es sie tatsächlich verwendet. Es wird eine Datendatei im aktuellen Verzeichnis mit dem Namen light_curve.dat erstellt, Siril zeichnet das Ergebnis in ein PNG-Bild, falls verfügbar

Die Ringradien für den Ring können entweder in den Einstellungen konfiguriert werden oder auf einen Faktor der FWHM des Referenzbildes gesetzt werden, wenn **-autoring** übergeben wird. Diese Autoring-Größen sind das 4,2-fache und das 6,3-fache der FWHM für den inneren bzw. äußeren Radius.

Siehe auch den Befehl **setphot**, um auf die gleiche Weise die Größe des Blendenradius einzustellen.

Siehe auch SEQPSF für Operationen an einem einzelnen Stern

Verweis: *seqpsf*

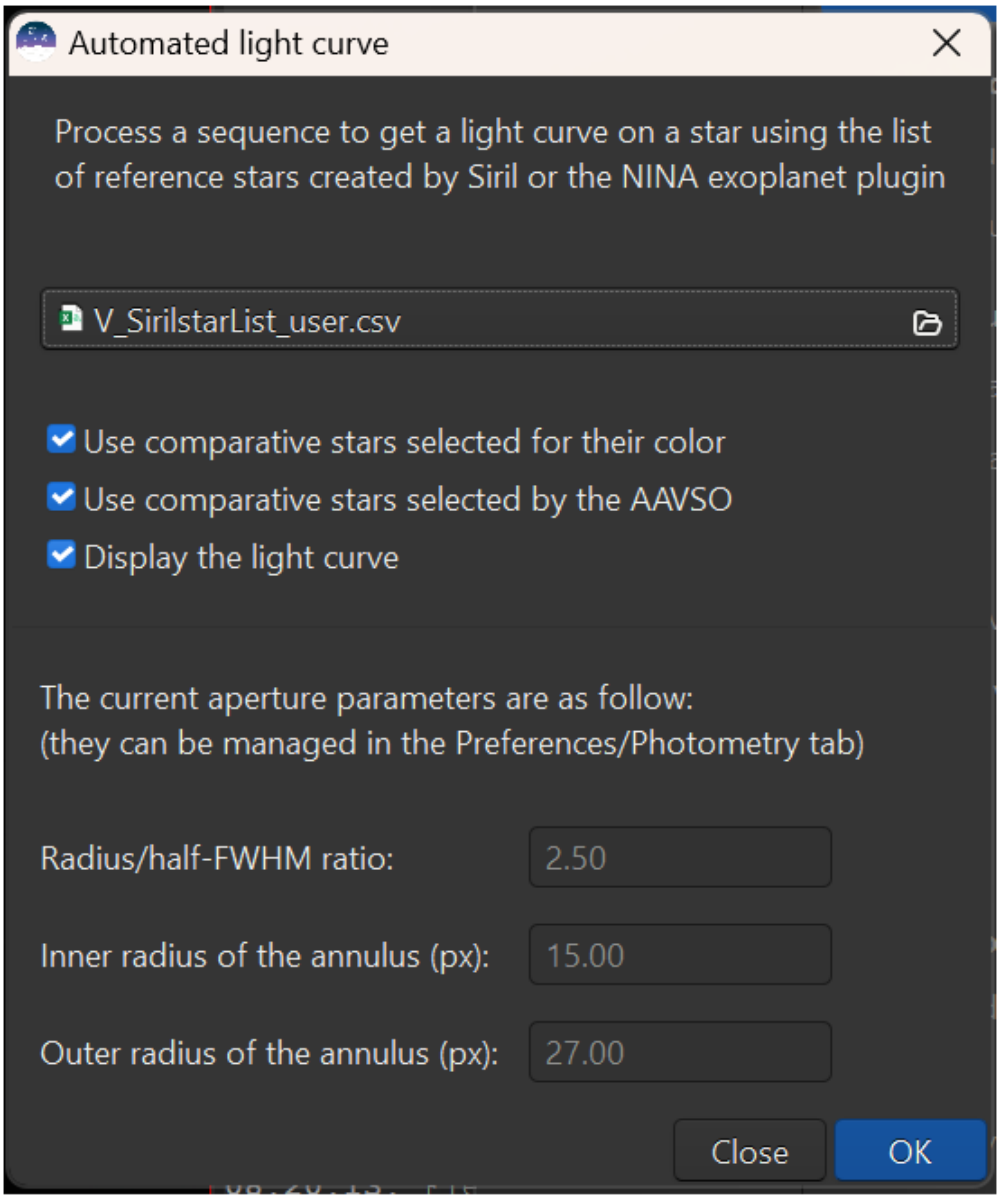


Abb. 16: Fenster "Automatische Photometrie".

16.3.3 Generieren der Liste der Vergleichssterne

Wenn Sie kein NINA-Benutzer sind, aber die *automatische Lichtkurvenfunktion* verwenden möchten, können Sie seit Siril-1.3 mit dem neuen Siril-Tool eine Liste von Vergleichssternen erstellen.

Liste über die grafische Benutzeroberfläche

Auf diese Funktion kann über das Menü *Extras*, *Photometrie* → *Vergleichssterndatei erstellen...* zugegriffen werden, sobald das aktuell angezeigte Bild der geladenen Sequenz astrometrisch gelöst wurde.

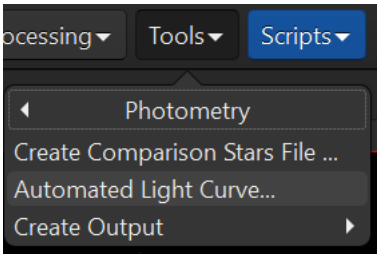


Abb. 17: Die Funktion *Vergleichssterndatei erstellen...* im *Extras*-Menü.

Voraussetzung ist das Laden eines repräsentativen, astrometrisch gelösten Bildes der Sequenz.

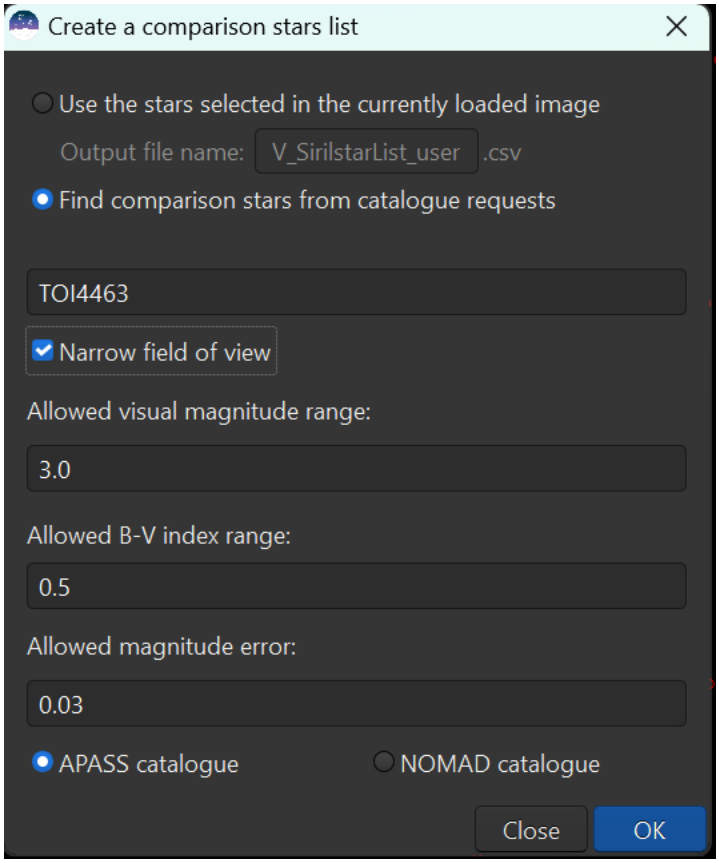
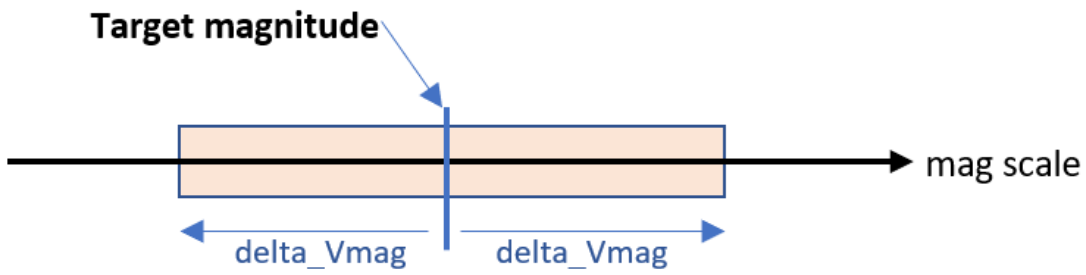
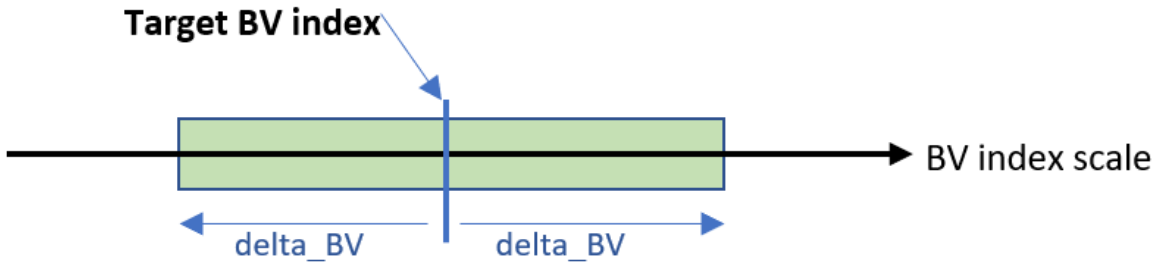


Abb. 18: Grundlegende Benutzeroberfläche für das Fenster „Vergleichssterne“.

- Die **Vergleichssterne aus Katalog-Anfrage finden** ist in diesem Anwendungsfall zwingend erforderlich.
- Das erste Feld muss der **Name des veränderlichen Sterns** sein, dessen Lichtkurve Sie erhalten möchten.

- Mit dem Kontrollkästchen **Kleines Bildfeld** können Sie einen Kreis mit der Mitte des Bildes auswählen.
- Das zweite Feld ist der **delta_Vmag**-Wert. Dies ist die maximale Abweichung vom Ziel **V-Magnitude** (0,0 < delta_Vmag < 6,0). Der Standardwert ist auf 3,0 eingestellt.
- Das dritte Feld ist der **delta_BV**-Wert. Dies ist die maximale Abweichung vom angestrebten **BV-Index** (0,0 < delta_BV < 0,7). Der Standardwert ist auf 0,5 eingestellt.
- Das vierte Feld ist der **max_emag**-Wert. Dies ist der maximal zulässige statistische Fehler bei Vamg (0,0 < max_emag < 0,1). Der Standardwert ist auf 0,03 eingestellt.
- Anschließend können Sie die Quelle der Vergleichssterne auswählen: aus dem APASS-Katalog oder dem NOMAD-Katalog.





Diese beiden Kriterien helfen Ihnen dabei, Vergleichssterne zu erhalten, die der Zielgröße und dem Farbindex entsprechen.

Die maximale Anzahl der Vergleichssterne ist intern auf 20 festgelegt.

Nachdem Sie auf *OK* geklickt haben, wird eine Anfrage an den entsprechenden Server gesendet.

Die Antwort ist eine Liste von Sternen, die aus dem ausgewählten Katalog stammen, Ihren Kriterien entsprechen, in Ihrem Bild rot angezeigt und als CSV-Datei gespeichert werden, die mit dem NINA-Exoplaneten-Prozess kompatibel ist.

Hinweis: Da die Sterne rot gekennzeichnet sind, gehören sie zu einem temporären Katalog, der geleert wird, nachdem die Schaltfläche „:guilabel:`Beschriftungen“ umgeschaltet wurde.

Tipp

Weitere Informationen zur Auswahl der Vergleichssterne finden Sie im „DSLR Observing Manual – Version 1.4 <https://www.aavso.org/sites/default/files/publications_files/dslr_manual/AAVSO_DSLR_Observing_Manual_V1-4.pdf>_ Seite 67 der AAVSO.

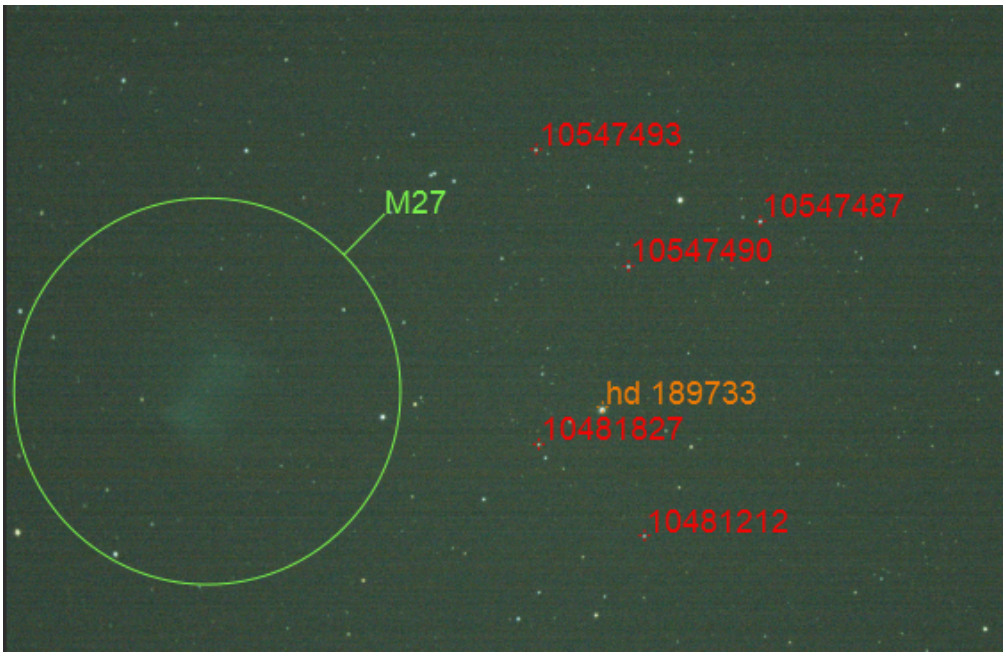


Abb. 19: Rendering-Beispiel nach einer APASS-Anfrage.

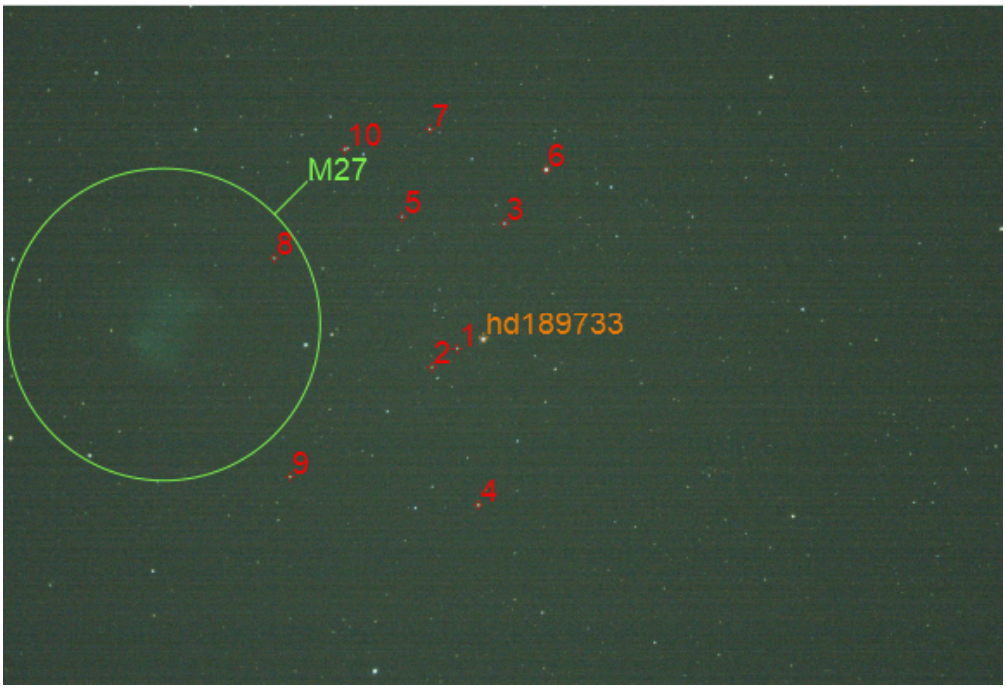


Abb. 20: Rendering-Beispiel nach einer NOMAD-Anfrage.

Liste durch Befehl

Es gibt auch einen Befehl zum Ausführen des Prozesses „Vergleichssterne finden“. Dadurch können Sie Ihre Auswahl noch weiter verfeinern.

Siril Kommandozeile

```
findcompstars star_name [-narrow|-wide] [-catalog={nomad|apass}] [-dvmag=3] [-dbv=0.
↪5] [-emag=0.03] [-out=nina_file.csv]
```

Findet automatisch Vergleichssterne im Feld des geladenen Bildes für die photometrische Analyse der Lichtkurve eines Sterne gemäß
- the provided name of the star
- the field of view of the image, reduced to a diameter of its height if **-narrow** is passed, avoiding stars in the corners
- the chosen catalog (APASS by default), can be changed with **-catalog={NOMAD|APASS}**
- the difference in visual magnitude from the variable star, in the range [0, 6] with a default of 3, changed with **-dvmag=**
- the difference in color with the variable star, in the range [0.0, 0.7] of their B-V indices with a default of 0.5, changed with **-dbv=**
- the maximum allowed error on Vmag in the range [0.0, 0.1] with a default of 0.03, changed with **-emag=**.

Die Liste kann optional als CSV-Datei gespeichert werden, die mit der NINA-Vergleichssternliste kompatibel ist, indem der Dateiname mit **-out=** angegeben wird. Ist der angegebene Name der spezielle Wert **auto**, wird er unter Verwendung der Eingabeparameter erzeugt

Siehe auch LIGHT_CURVE

Verweise: *light_curve*

- `-narrow` begrenzt das Suchfeld auf einen Kreis, der in Ihr Bild eingeschrieben und zentriert ist. Dadurch wird vermieden, dass Sterne in einem möglicherweise verzerrten Teil des Bildes ausgewählt werden.
- Ohne die Option „-narrow“ wird nach Vergleichssternen im gesamten Bild gesucht. Dies kann bei einem Sichtfeld mit wenigen Sternen nützlich sein.
- Mit „[-catalog={nomad|apass}]“ können Sie zwischen dem NOMAD-Katalog (default) und dem APASS-Katalog wählen.

Tipp

Die angezeigten Namen werden auf einfache Zahlen reduziert und nach zunehmendem Abstand zur Bildmitte sortiert. Diese Beschriftungen dienen nur der Information.

- `[-dvmag=3]`, `[-dbv=0.5]` und `[-max_emag=0.03]` stehen jeweils für die **delta_Vmag**, **delta_BV** und **max_emag**-Kriterien wie zuvor erläutert.
- `[-out=nina_file.csv]` setzt den Namen der CSV-Datei.

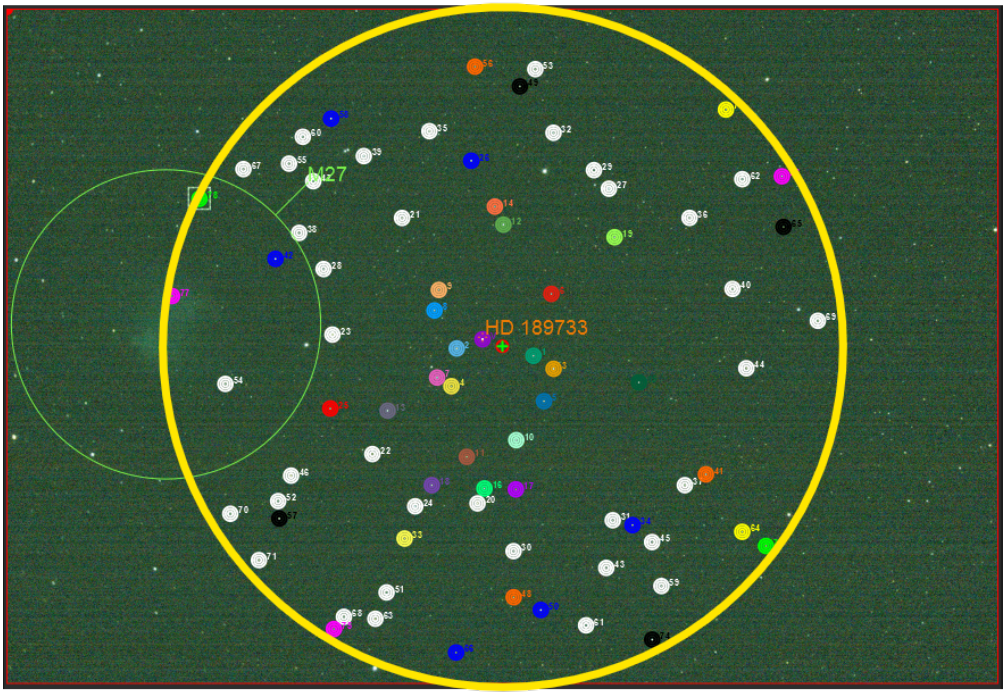


Abb. 21: Ansicht des durch „-narrow“ eingeschränkten Suchfelds. (Der gelbe Kreis dient nur zur Veranschaulichung)

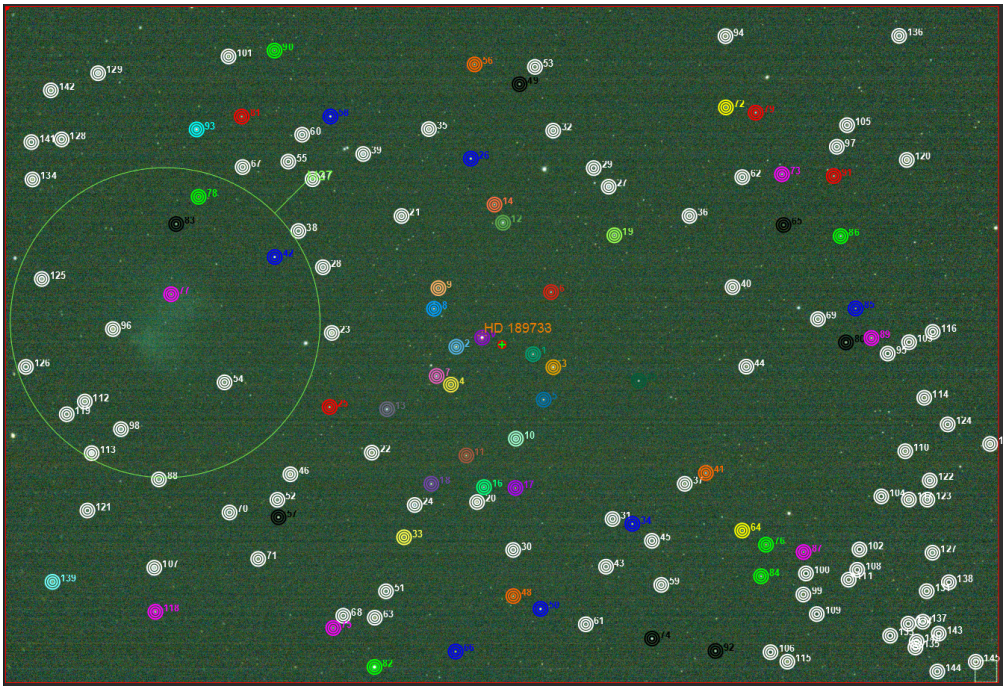


Abb. 22: Ansicht eines größeren Suchfeldes.

Tipp

APASS steht für AAVSO Photometric All Sky Survey. Weitere Informationen finden Sie hier.

Di rAAVSO Photometric All-Sky Survey (APASS DR9) liefert kalibrierte Helligkeiten im Bereich 7,0 < Vmag < 17,0 für den gesamten Himmel in den BVugriZsY Bändern.

Die Photometrie ist auf etwa 0,02 mag genau, und die Astrometrie ist auf etwa 0,15 Bogensekunden genau (siehe „Verwendung von APASS und 2GSS zur Untersuchung veränderlicher Sterne <https://www.researchgate.net/publication/319603237_Using_APASS_and_2GSS_for_studying_variable_stars/fulltext/59b4be50a6fdcc3f8895a420/U“) singen- APASS-and-2GSS-for-studying-variable-stars.pdf>`_).

Verwerfen potenzieller veränderlicher Sterne

Unter den von APASS oder NOMAD zurückgegebenen Sternen können einige als veränderliche Sterne identifiziert werden. Wenn einer (oder mehrere) dieser Sterne als Vergleichssterne ausgewählt werden, kann ihre Flussschwankung das Endergebnis für den untersuchten veränderlichen Stern beeinflussen.

Die Funktion zum automatischen Sternenlistenvergleich von Siril ist also in der Lage, diese Ausreißersterne zu finden und sie entsprechend zu verwerfen.

Zur Erkennung veränderlicher Sterne können drei Hauptkataloge verwendet werden:

- Der Allgemeine Katalog variabler Sterne (GCVS). Bietet mehr als 89.000 veränderliche Sterne.
- Der AAVSO VSX listet alle Sterne auf, die einen Eintrag im AAVSO International Variable Star Index haben.
- Das GAIA varisum GAIA varisum, eine Untertabelle von GAIA DR3, die „potenzielle variable Objekte“ auflistet.

Diese 3 Kataloge können während des Verwerfungsprozesses einzeln oder alle gleichzeitig verwendet werden. Allerdings muss man sich darüber im Klaren sein, dass jeder ausgewählte Katalog eine zusätzliche Online-Anfrage nach sich zieht.

Grundsätzlich wird nur das GAIA Varisum ausgewählt. Der Benutzer kann jedoch weiterhin seine eigene Wahl treffen.

Die verwendeten Kataloge werden mit einer Siril-Variablen `photometry.discard_var_catalogues` gemäß der folgenden Tabelle verwaltet:

Tab. 1: Verwerfen Sie die Tabelle, um `photometry.discard_var_catalogues` richtig festzulegen.

Variablenwert	Binär	GCVS	AAVSO VSX	GAIA Varisum
0	0000			
1	0001	✓		
2	0010		✓	
3	0011	✓	✓	
4	0100			✓
5	0101	✓		✓
6	0110		✓	✓
7	0111	✓	✓	✓

(Aus der vorherigen Tabelle ist ersichtlich, dass die Variable im Binärformat geschrieben werden kann, mit „Bit#0“ als GCVS, „Bit#1“ als VSX und „Bit#2“ als GAIA- Varisum).

Diese Variable kann man mit dem folgenden Befehl lesen:

```
get photometry.discard_var_catalogues
```

Mit dem folgenden Befehl kann man der Variablen einen anderen Wert zuweisen:

```
set photometry.discard_var_catalogues=4
```

Als Beispiel mit dem Befehl:

```
set photometry.discard_var_catalogues=7
```

Sie haben die 3 Kataloge angekreuzt und in der Konsole wird angezeigt:

```
Contacting server
-> 5 variable stars found within the image from AAVSO Variable stars
Contacting server
-> 1 variable stars found within the image from GCVS
Contacting server
-> 8 variable stars found within the image from Gaia DR3 Variability
```

Struktur der Ausgabedatei

Wenn Sie den Befehl „findcompstars“ und den Parameter``-out=auto`` oder die GUI-Schaltfläche *Comparison stars festlegen* verwenden, erhalten Sie einen Dateinamen mit der folgenden Struktur:

`[star_name]_SirilstarList_[delta_Vmag]_[delta_BV]_[max_emag]_[catalogue].csv`

Zusätzliche Informationen sind im Datei-Header (als Kommentar) verfügbar:

```
# Sorted comparison stars for KELT-16 from APASS according to the following criteria
# Siril: 12 stars, dVmag 1.00, dBV 0.30, max e_mag 0.03
type,name,ra,dec,mag
Target,KELT-16,314.268494,31.661009,11.72
Comp1,1,314.256578,31.660406,12.307
Comp1,2,314.317434,31.754368,12.431
Comp1,3,314.305316,31.546814,11.682
Comp1,4,314.263680,31.541927,12.084
Comp1,5,314.465451,31.626255,12.084
Comp1,6,314.459388,31.596876,11.979
Comp1,7,314.452316,31.746360,11.504
Comp1,8,314.437181,31.827260,11.24
Comp1,9,314.314246,31.879192,12.313
Comp1,10,314.431752,31.839763,10.88
Comp1,11,314.006569,31.666214,11.896
Comp1,12,314.304606,31.925116,11.713
```

KAPITEL 17

Plot-Funktion

Siril verfügt über eine Registerkarte, auf der die während der Registrierung oder anderer Berechnungen ermittelten Daten graphisch dargestellt werden. Diese Registerkarte ist sehr leistungsfähig und ermöglich ein einfaches Sortieren der Bilder sowie eine eingehende Analyse derselben. Das Tastenkürzel für den Zugriff auf diese Registerkarte lautet `F5`.

17.1 Plot/Grafische Darstellung der Registrierungsdaten

Um die manuelle Sortierung und Auswahl Ihrer registrierten Einzelbilder zu verbessern, wurden auf der Registerkarte "Plot" Möglichkeiten zur grafischen Darstellung hinzugefügt. Nach Abschluss der Registrierung Ihrer Sequenz (oder beim Laden einer registrierten Sequenz) können Sie nun in der Dropdown-Liste die Parameter auswählen, die für die Darstellung und Sortierung Ihrer Daten von Interesse sind.

Sie können außerdem wählen, ob Sie einen der Parameter *gegen* einen Anderen darstellen möchten. In der Dropdown-Liste sind folgende Optionen verfügbar:

- **FWHM**: Dies ist die maximale Breite bei halbem Maximum, eines der gängigsten Kriterien zur Beurteilung der Qualität eines Deep-Sky-Bildes.
- **Rundheit**: Die Rundheit r wird als Verhältnis $\frac{\mathrm{FWHMy}}{\mathrm{FWHMx}}$ berechnet.
- **wFWHM**: Dies ist eine Verbesserung der einfachen FWHM. Die FWHM wird nach der Anzahl der Sterne im Bild gewichtet. Bei gleicher FWHM-Messung hat ein Bild mit mehr Sternen ein besseres wFWHM als ein Bild mit weniger Sternen. Auf diese Weise können viel mehr schlechte Bilder ausgeschlossen werden, indem die Anzahl der erkannten Sterne im Vergleich zum Referenzbild verwendet wird.
- **Hintergrund**: Durchschnittlicher Wert des Himmelshintergrundes.
- **# Sterne**: Dies ist die Anzahl der Sterne welche für die Registrierung genutzt wurden.
- **X-Position**: Verschiebung in X-Richtung bezogen auf das Referenzbild.
- **Y-Position**: Verschiebung in Y-Richtung bezogen auf das Referenzbild.
- **Qualität**: Dieses Kriterium ist eine Zahl im Bereich [0, 1], welche die Qualität angibt, die mit einem beliebigen Planeten-Registrierungsalgorithmus verarbeitet wurden.

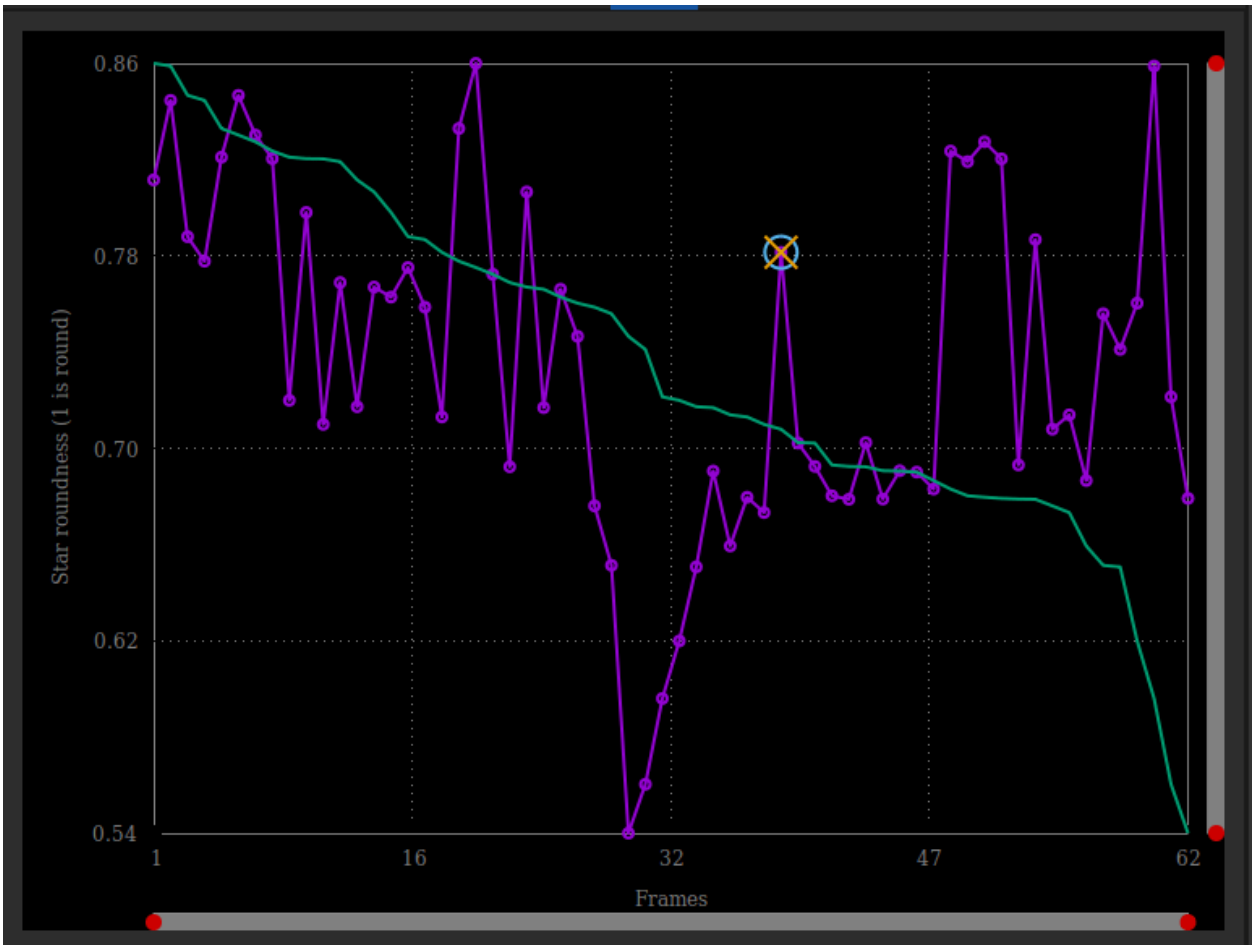


Abb. 1: Registerkarte *Plot* wie sie nach einer globalen Registrierung angezeigt wird.

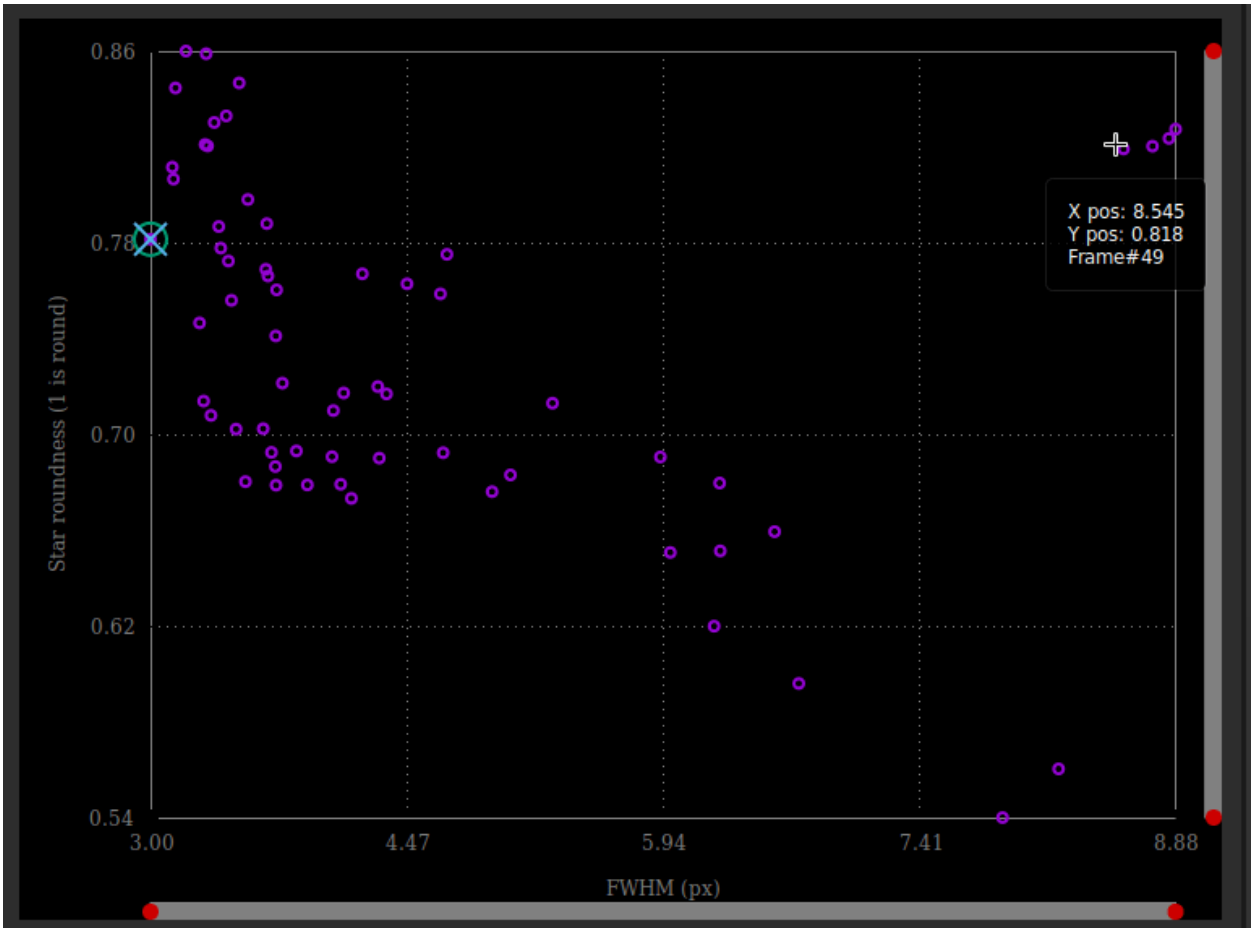


Abb. 2: Die Werte von Rundheit und FWHM werden als Punktwolke angezeigt. Bewegen Sie den Mauszeiger über die verschiedenen Datenpunkte, um die X- und Y-Werte zusammen mit der entsprechenden Bildnummer anzuzeigen.

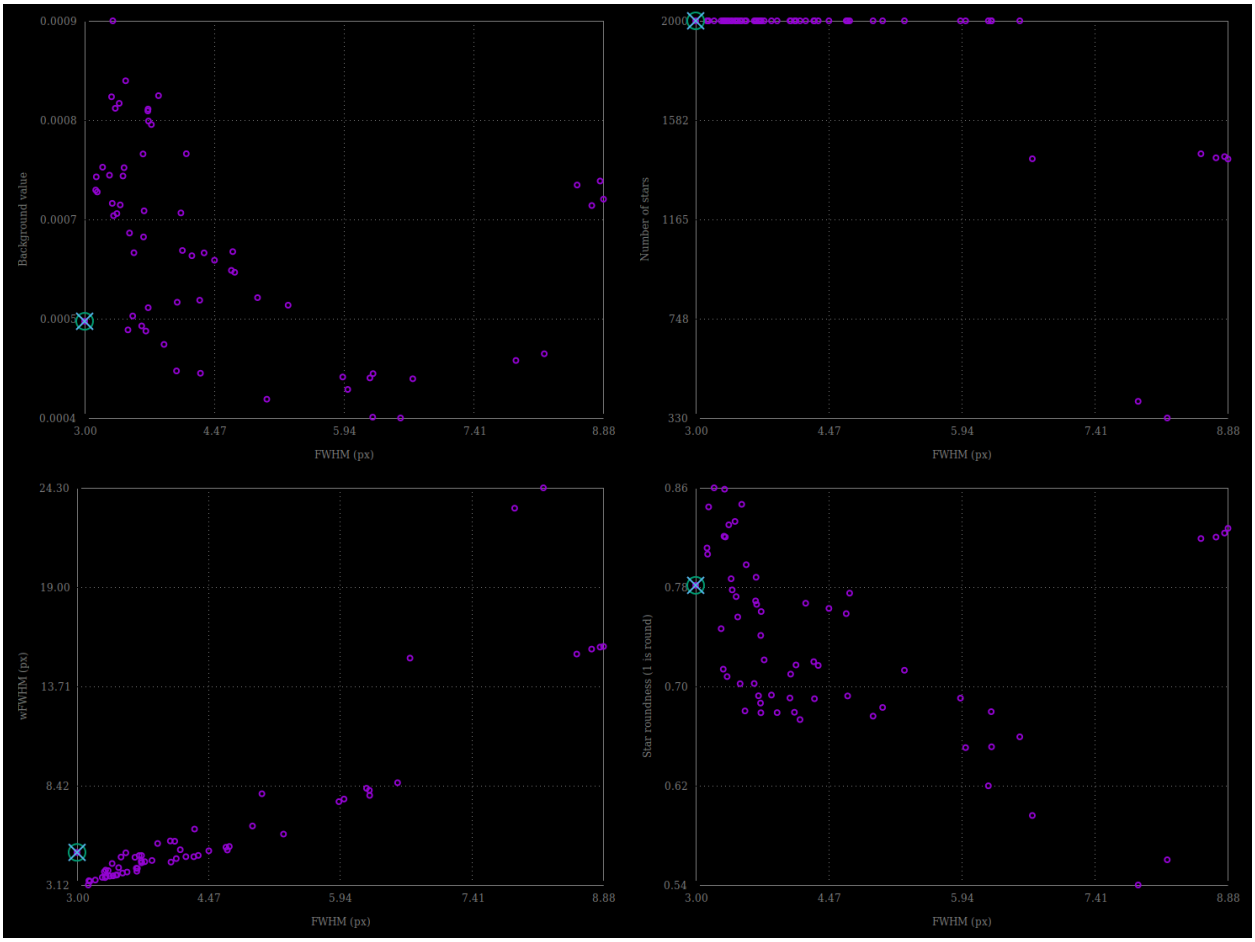


Abb. 3: Unterschiedliche Möglichkeiten der grafischen Darstellung mit dem selben Satz aufgenommener Bilder.

Klicken sie auf einen der Datenpunke, um das Einzelbild auszuschließen oder zu öffnen. Mit der zweiten Option wird das Bild geladen und die Frameauswahl eingeblendet. Der für die Y-Werte gewählte Parameter wird in der letzten Spalte der Bildauswahl angezeigt, die dann zum Sortieren, Überprüfen oder Auswählen bzw. Abwählen von Einzelframes aus der Sequenz verwendet werden kann.

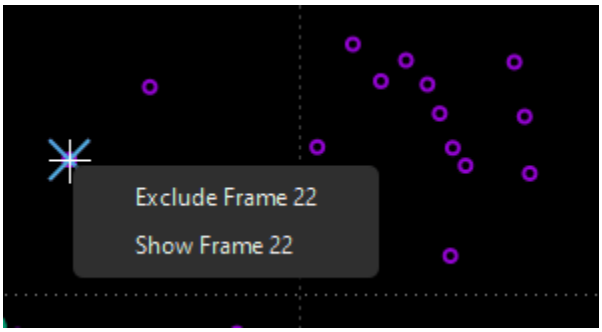


Abb. 4: Klicken Sie mit der rechten Maustaste auf einen Datenpunkt, um ihn auszuschließen oder in die Bildvorschau zu laden.

Sie können auch mehrere Datenpunkte auswählen bzw. deren Auswahl aufheben, indem Sie eine Auswahl im Diagramm zeichnen. Die Informationen am oberen Rand der Auswahl geben die Anzahl der ausgewählten Punkte sowie die Grenzwerte der Auswahl an. Sie können die Auswahl genauso umgestalten wie eine gezeichnete Auswahl in der Bildansicht. Wenn Sie mit Ihrer Auswahl zufrieden sind, können Sie mit einem Rechtsklick ein Menü aufrufen, in dem Sie die Punkte beibehalten oder ausschließen oder den Zoom auf die Auswahl einstellen können.

17.2 Photometrische Grafik

Als Ergänzung zum Sortieren und Filtern von Bildern einer Sequenz können Sie auch eine PSF eines Sterns für die gesamte Sequenz durchführen. Das Verfahren wird auf der Seite *photometry* detailliert beschrieben. Dann wird der Photometrie-Eintrag in der ersten Dropdown-Liste aktiviert und automatisch ausgewählt. Die übrigen Dropdown-Listen enthalten die folgenden Einträge:

- **FWHM**: Maximale Helligkeit bei halbem Maximum, wie oben definiert.
- **Rundheit**: Die Rundheit r wird als Verhältnis $\frac{\text{FWHMy}}{\text{FWHMx}}$ berechnet.
- **Amplitude**: Dies ist der maximalwert der angepassten Funktion, der sich an den Schwerpunktkoordinate befindet.
- **Magnitude**: Relative Magnitude/Helligkeit des analysierten Sterns.
- **Hintergrund**: Durchschnitt der lokalen Hintergrundhelligkeit im Bereich der PSF.
- **X-Position**: Verschiebung in X-Richtung bezogen auf das Referenzbild.
- **Y-Position**: Verschiebung in Y-Richtung bezogen auf das Referenzbild.
- **SNR**: Eine Abschätzung des Signal-Rauschverhältnisses.

Bei der Photometrie ist es im Gegensatz zur Registrierung nicht möglich, die X-Achse zu ändern. Und nur die Anzahl der Bilder kann verwendet werden (oder der julianische Tag).

- Die Schaltfläche *Plots löschen* kombiniert die Menüs *Alle löschen* und *Letzte löschen*, um entweder alle photometrischen Kurven oder die zuletzt gezeichnete Kurve zu löschen.

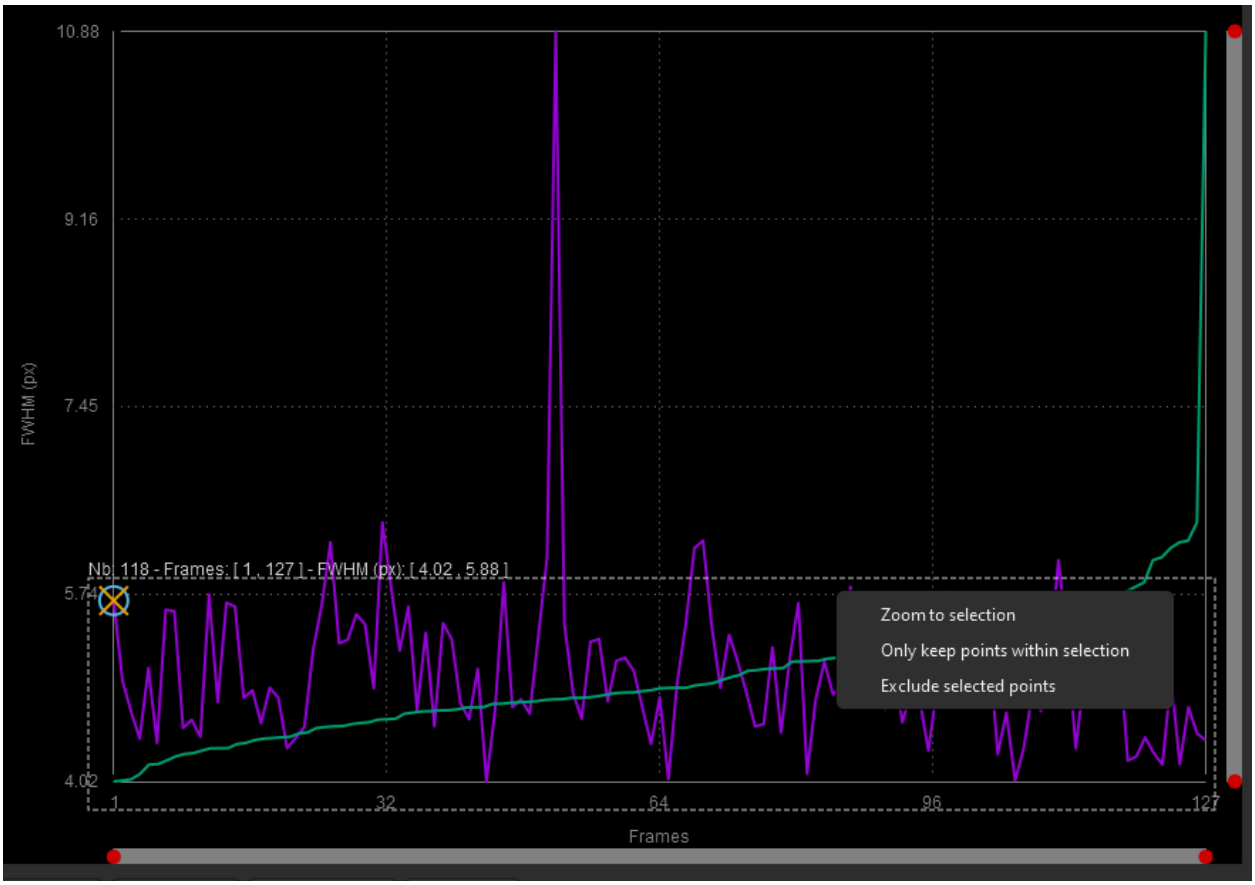


Abb. 5: Klicken Sie mit der rechten Maustaste auf eine gezeichnete Auswahl, um eine Massenauswahl zu treffen bzw. die Auswahl aufzuheben oder um zu zoomen.

17.2.1 Werkzeugmenü

Ab Siril 1.4 wurden viele Optionen in das Menü *Werkzeuge* verschoben, das sich rechts neben dem Menü *Bildbearbeitung* befindet. Dieses Menü wird im Abschnitt *Photometrie* näher erläutert.

17.3 Gemeinsame Optionen

Mehrere Ausgabedateien können durch Klicken auf die Schaltfläche *Ausgabe* erzeugt werden:

- *Gültige Daten in Siril-Plot anzeigen* ist eine Option, die alles im Plot-Tab in einer neuen Instanz von Siril-Plot anzeigt.
- Die Schaltfläche *Exportieren als CSV* exportiert das angezeigte Diagramm in eine CSV-Datei.

Die Schaltfläche *Anzeige* dient zum Ändern der Achseneinheiten. Sie enthält, je nach Verfügbarkeit, mehrere Schaltflächen in einem Dropdown-Menü:

- Wenn die Abtastrate des Bildes bekannt ist, zeigt die Schaltfläche *Bodensekunde* die FWHM in Bogensekunden statt in Pixeln an.
- Im Falle der reinen Photometrie (also nicht wirklich eine gemeinsame Option) ist es auch möglich, *Julianisches Datum* als Einheit für die x-Achse zu wählen.

Tipp

Wenn Sie mit dem Mauszeiger über den Text "Überfahren für Legende" fahren, wird die Legende des Plots angezeigt.

- Violette Kurve: Tatsächliche Darstellung, wie sie mit den X- und Y-Auswahlen konfiguriert wurde.
- Grüne Kurve: Sortierte Werte in der Reihenfolge abnehmender Qualität.
- Kreismarkierung: Der Wert für das Referenzbild.
- Kreuzmarkierung: Der Wert für das aktuell geladene Bild.

17.4 Grafik-Interaktionen

Hier ist eine Zusammenfassung der möglichen Interaktionen mit dem Grafikfenster:

- **Links-Klick** in einen Schieberegler: Setzt den nächsten roten Punkt darauf
- **Doppelklick** auf einen Schieberegler: Setzt diese Achse zurück
- **Rechts-Klick + verschieben** auf einem Schieberegler : Verschiebt den angezeigten Bereich
- **Links-Klick + verschieben** auf einem Schieberegler: Zeichnet eine Auswahl
- **Link-Klick + verschieben** auf dem Rand der Auswahl : Verändert die Größe der Auswahl
- **Doppelklick** in der Grafik : Setzt den Zoom beider Achsen zurück
- **Rechts-Klick** wenn eine Auswahl aktiv ist : Zeigt das Menü für : Zoom auf die Auswahl / Behalte nur die Punkte der Auswahl / Schließe die Punkte der Auswahl aus
- **Links-Klick** wenn eine Auswahl aktiv ist : Lösche die Auswahl

KAPITEL 18

Statistik

Dies ist eine Dokumentation für die Siril-Statistiken, die über die grafische Benutzeroberfläche (GUI) aus dem Menü *Werkzeuge→ Bildanalyse*) und der anschließenden Auswahl von *Statistik...* oder über den Befehl stat genutzt werden können. Beachten Sie, dass es bei der Verwendung der grafischen Benutzeroberfläche möglich ist, eine Auswahl in das geladene Bild zu zeichnen, und dass dabei die Statistik nur mit den Pixeln der Region berechnet wird.

Mit der Option *Pro CFA-Kanal* können Sie Statistiken für jeden R-, G- und B-Kanal in CFA-Bildern/Farbbildern berechnen, auch wenn das Bild nicht debayert wurde.

Tipp

When a CFA image is loaded, the selection will be constrained to a minimum size of 2x2 for Bayer images and 3x3 for X-Trans images so that there is at least one pixel in each Bayer channel within the selection, to ensure that it is possible to generate statistics.

Viele dieser Werte sind Maßzahlen für das Streuungsmaß.

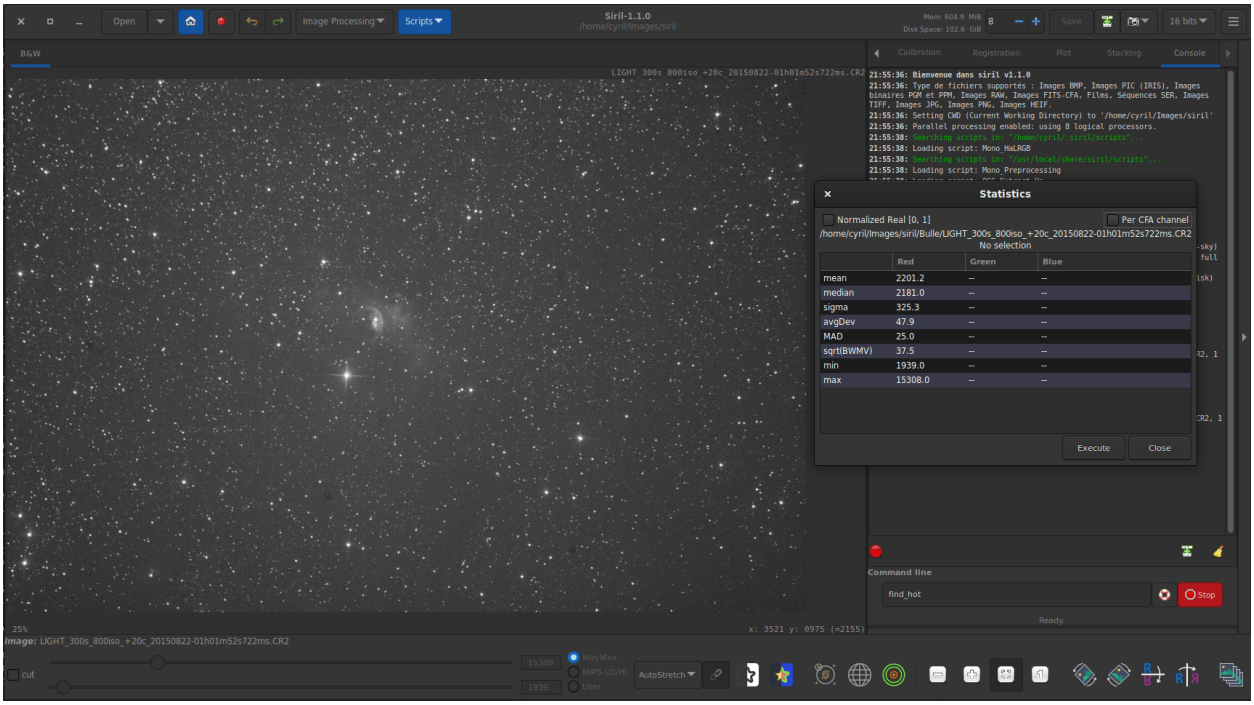


Abb. 1: Ein-Kanal-Statistik für ein CFA-/Farbbild. Die angegebenen Werte sind in diesem Fall nicht wirklich relevant.

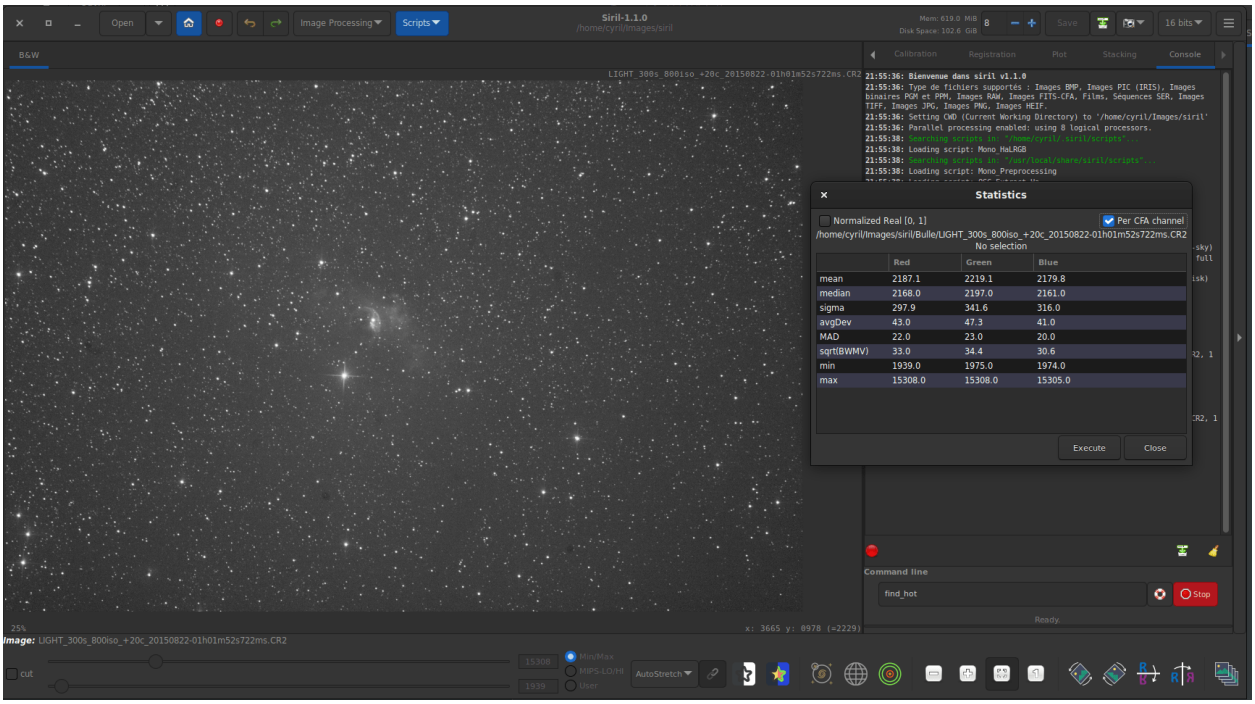


Abb. 2: Drei-Kanal-Statistik für ein CFA-/Farbbild.

Siril Kommandozeile

```
stat [-cfa] [main]
```

Liefert Statistiken für das aktuelle Bild, standardmäßig die Basiswerte oder die Hauptliste, wenn **main** übergeben wird. Wenn eine Auswahl getroffen wird, werden die Statistiken innerhalb der Auswahl berechnet. Wenn **-cfa** übergeben wird und es sich um ein CFA-Bild handelt, werden die Statistiken für die einzelnen Farbkanäle erstellt

18.1 Abschätzungen

18.1.1 Mittelwert

Dies ist der arithmetische Mittelwert, auch Durchschnitt oder arithmetisches Mittel genannt. Er wird berechnet, indem die Summe der Pixelwerte durch deren Anzahl in einem Bildkanal geteilt wird.

18.1.2 Median

Der Median, oder Zentralwert ist der Wert, der die obere Hälfte von der unteren Hälfte eines Datensatzes trennt. Im Allgemeinen stellt er den Wert des Hintergrundes einer astronomischen Aufnahme dar.

18.1.3 Sigma

Auch bekannt als Standardabweichung oder Varianz oder $sigma$, ist ein Maß für die Streuung der Bildpixel, basierend auf den quadrierten Differenzen zum Durchschnitt. Der Sigma-Wert eines Teilbildes, das nur den Hintergrund enthält, stellt das Rauschen des Bildes dar.

18.1.4 Hintergrundrauschen

Diese Abschätzung ist in der grafischen Benutzeroberfläche unter dem Menüpunkt *Tools* → *Image Analysis* → *Noise estimation* verfügbar und wird auch am Ende des Stackings angezeigt.

Dies ist ein Maß für das geschätzte Rauschen im Bildhintergrund, für Pixel, die einen Wert haben, der niedrig genug ist, um als Hintergrund zu gelten. Es handelt sich um einen iterativen Prozess auf der Grundlage von k.sigma (ein Faktor der Standardabweichung über dem Median), so dass es keinen festen Schwellenwert für "niedrig genug" gibt.

Siril Kommandozeile

```
bgnoise
```

Gibt den Rauschwert des aktuell geladenen Bildes aus

18.1.5 avgDev

Die durchschnittliche Abweichung, auch AAD für average absolute deviation oder durchschnittliche absolute Abweichung. Um zu verstehen, was die durchschnittliche Abweichung bedeutet, muss man wissen, was der Begriff der absoluten Abweichung ist. Die absolute Abweichung ist der Abstand zwischen jedem Wert in einem Datensatz und dem Mittelwert (in diesem Fall) oder dem Median (für MAD unten) dieses Datensatzes. Nimmt man alle diese absoluten Abweichungen und ermittelt den Durchschnitt, so erhält man die mittlere durchschnittliche Abweichung. Zur Vereinfachung: Wenn die Standardabweichung die quadrierte Abweichung vom Mittelwert ist, ist die AAD die lineare Version davon.

18.1.6 MAD

Die Mittlere absolute Abweichung ist ein robustes Maß für die Streuung eines Datensatzes. Die absolute Abweichung und die Standardabweichung sind ebenfalls maßzahlen für die Streuung, werden aber stärker von extrem hohen oder extrem niedrigen Werten beeinflusst. Sie ähnelt der obigen durchschnittlichen Abweichung, bezieht sich aber auf den Median und nicht auf den Mittelwert.

18.1.7 BWMV

Die gewichtete mittlere Varianz biweight midvariance ist ein weiteres Instrument zur Messung der Streuung eines Datensatzes, das noch robuster gegenüber Ausreißern ist als die anderen oben erwähnten Werte. Es verwirft Datenpunkte, die zu weit vom Median entfernt sind, und berechnet eine gewichtete Abweichung, wobei die Gewichte abnehmen, je weiter die Datenpunkte vom Median entfernt sind. Die Abschätzung der Streuung ist die Quadratwurzel (gekennzeichnet als $\sqrt{BWMV}$) dieses Wertes.

18.1.8 Lage-Skalen-Familie

Diese Parameter, umgangssprachlich oft Skalierung und Offset genannt, werden in den Benutzeroberflächen nicht angezeigt, sondern von Siril intern berechnet. Um die Histogramme der verschiedenen Bilder für die Normalisierung vor dem Stacking auszurichten, muss man berechnen, wo sie in Bezug auf die Höhe liegen und wie breit sie in Bezug auf die Streuung sind. Als gültiger Schätzwert für die Lage könnte der Median herangezogen werden, während der MAD oder das $\sqrt{BWMV}$ für die Skalierung verwendet werden könnte. Um jedoch die Robustheit der Werte zu erhöhen, werden die Pixel, die mehr als das $6 \times$ MAD vom Median entfernt sind, verworfen. Auf diesem beschnittenen Datensatz werden der Median und $\sqrt{BWMV}$ neu berechnet und als Standort- bzw. Skalenschätzer verwendet. Sie werden relativ zum Referenzbild einer Sequenz in Siril berechnet.

KAPITEL 19

Siril für Wissenschaftler

For years Siril has been a great tool for amateur astronomers. With its capability to produce light curves, query online catalogues, run operations on image sequences, and many other recent additions, it has also become a great tool for scientific analysis. This page is a quick reference for scientists who want to learn quickly about Siril, in form of a list of Q&A.

19.1 How can Siril help me with my research?

Siril's default file format is FITS files, containing either a single image or several (see the FITS cubes question below). Siril has always been developed with data accuracy in mind. Displaying an image can be done in various ways without destroying input data.

Siril has many tools that can help scientific observations: star detection, elaborate plate solving, registration of an image sequence using stars or using astrometric solutions, aperture photometry with magnitude calibration using catalogues, automated light curve creation, object lookup from online catalogues (stars, galaxies, solar system objects). Many of those operations can be done from the GUI or from text inputs (commands or scripts), some operations produce comma-separated values (CSV) files for results, some can produce graphs directly in Siril.

19.2 Which version of Siril should I install?

Siril follows an even/odd versioning system, e.g. 1.2.x was a stable series whereas 1.3.x was a development series. For scientific use it is recommended to install the latest version of the latest stable series: download links can be found at siril.org. Beta versions, those distributed on the website, can also be considered quite stable for general use.

19.3 How can I get in touch with the developers?

The official forum is on pixls.us. If you want to contact us privately you can send us a direct message there or on gitlab. Our email addresses can also be found in the AUTHORS file. If you speak French you can find us on Astrosurf or Webastro. Feel free to reach out if you have specific needs for your research.

19.4 How accurate are Siril results?

Many algorithms in Siril rely on peer-reviewed published algorithms which can be found in the documentation page of the features. We are testing Siril often, sometimes comparing results with other tools, but we don't have the resources to have a automated tests on all pieces of the software, so data comes with no warranty. We know some things could be improved, for example error computation on magnitudes, which depend on camera gain, not always available in the FITS header.

Astrometric data is as good as any and supports the WCS FITS convention.

19.5 How can I cite Siril?

If you use Siril in your research, please cite:

> Richard, C., et al. (2024). Siril: An Advanced Tool for Astronomical Image Processing. *Journal of Open Source Software*, 9(102), 7242. https://doi.org/10.21105/joss.07242

BibTeX format:

```
@article{Richard2024,
  title={Siril: An Advanced Tool for Astronomical Image Processing},
  author={Richard, Cyril and others},
  journal={Journal of Open Source Software},
  volume={9},
  number={102},
  pages={7242},
  year={2024},
  publisher={The Open Journal},
  doi={10.21105/joss.07242}
}
```

19.6 How can I open FITS cubes with Siril?

FITS cubes come in two shapes for Siril:

- those that contain several images of the same size and format, we call those FITS sequences or fitseq for short, and use them as an alternate file representation of image sequences;
- those that contain several images of changing properties, or even tables and other data formats. Siril will only read image data, but is able to work on each image of such FITS cubes if an option is enabled in settings (named *Allow FITS cubes to have images of different sizes* in *FITS Options*). Most sequence operations will not be available on this kind of file.

Siril can also extract images from FITS cubes to individual FITS files or to other formats if needed.

19.7 How do I measure FWHM in an image?

Siril can give you the FWHM of a single star if you draw a selection around it in the GUI, then right click and select PSF, or for the stars in an area of the image or the full image. You can also use the *quick PSF* button in the main toolbar.

More generally: open your image and open the *Dynamic PSF tool* (from the *Tools* menu). From there clicking on the first button will start the star detection, within the selected area if you drew one. Clicking on the Sigma icon will give mean values for all detected stars. See also the next question.

19.8 How do I use star detection (source extraction)?

Siril's source extractor can be found in the GUI in the *Dynamic PSF tool*, it is also available as the *findstar* command. It is normally only able to detect stars and will struggle if two stars are too close from each other, or if stars have asymmetrical or very non-Gaussian shapes, but there are many settings that you could adjust to your needs, including brightness, amplitude, roundness thresholds or star shape (Gaussian or Moffat).

The GUI will display the found stars by clicking on the button on the left of the lower bar, giving quick feedback on the detection settings. The GUI and the command can produce a CSV file that can then be used to check extracted sources in an image or in a sequence of images (command only).

The list of stars can be sorted by property, clicking on a row will highlight the star in the image.

19.9 Welche Astrometrie-Tools stehen zur Verfügung?

Siril has now a very high quality astrometric solver (plate solver) and tools that interact with the astrometric solution. All is based on the FITS WCS convention and on WCSLIB. Here is a list of tools related to astrometry:

- Plate solving with local Gaia DR3 catalogue extract or with remote full Gaia DR3, PPMXL or APASS.
- Display stars from the catalogue on image, to have a visual feedback of the accuracy of the astrometric solution, using the *conesearch* command.
- *Query* online catalogues for an object using its name and the image time.
- Get centroid equatorial J2000 coordinates for any detected object.
- Show a mark on image for user provided J2000 coordinates (*show* command).
- Align images on the celestial grid, possibly creating a mosaic (an image bigger than the original).
- Correct for geometrical aberration of images, using a master correction image based on astrometry.

19.10 Kann es Objekte des Sonnensystems verarbeiten?

Siril can query IMCCE services to find a solar system object coordinates and display the expected position on images. It can also display the known solar system objects in an image. Observer location will be required for those uses and can be configured with MPC codes in the settings. See the *annotations* documentation page.

Soon, Siril will be able to do synthetic tracking of known solar system objects, making them appear on stacking if they are too faint to be seen on single exposures.

But Siril is not made to process closeup view of planets with multi-point stacking like other tools such as Planetary-SystemStacker (PSS) does.

19.11 Kann Siril mit Spektren arbeiten?

Siril was not designed to handle spectrometry data. An *Intensitätsprofilierung* tool that shows the graph of pixel intensities can be used for preview of a spectrum. FITS tables are not displayed in any way.

19.12 Can Siril replace IRAF?

Some features will be missing, many will be much more easy to do with Siril. We don't yet have a list.

Siril has been developed since 2012 and we don't plan to stop anytime soon. There are 3 active developers and thousands of amateur users.

KAPITEL 20

Skripting

This section explains the different scripting and automation methods available in Siril.

Tipp

Python scripting was introduced in the 1.3.5 development version. It is currently marked as **EXPERIMENTAL**. This doesn't mean it will eat your data: the interface itself is robust and has been tested through development: the present experimental nature of it is more to do with the fact that we don't yet know what users will do with this new capability and whether there may be issues or limitations that we have not foreseen, perhaps due to the constraints of packaging or consistency across the different operating systems.

Please try it out, either as a user by using scripts published on the scripts repository, or as a script writer. We welcome all your feedback and will aim to refine the interface throughout the 1.4 stable series and 1.5 development series.

If you want to debug your Python scripts at runtime, tick the box next to *Enable Python debug mode*. A tutorial detailing the steps to attach to the Python process is shown in Siril tutorial page.

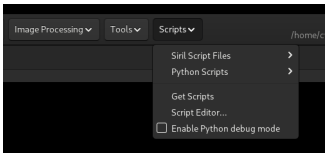


Abb. 1: Skriptmenü

20.1 SIRIL Skriptdateien

Siril Script Files are the original format for scripting Siril and remain a simple and useful way of automating a fixed set of tasks.

20.1.1 Siril Skripte

Siril-Skripte sind eine Liste von Befehlen, die entweder über die grafische Benutzeroberfläche oder über die Befehlszeilenschnittstelle ausgeführt werden können. Im Allgemeinen wirken Befehle, die ein einzelnes Bild verändern, auf das aktuell geladene Bild, sodass in Skripten der Befehl *load* verwendet werden muss, während Befehle, die auf eine Bildsequenz wirken, den Namen der Sequenz als Argument benötigen. Wenn Dateien nicht so benannt sind, dass Siril sie als Sequenz erkennt, hilft der Befehl *convert*.

> **Tipp**
>
> Das Zeichen `Space` ist das Begrenzungszeichen zwischen den Argumenten. Wenn Sie Leerzeichen innerhalb der Argumente benötigen, können Sie einfache oder doppelte Anführungszeichen verwenden, genau wie in der Befehlszeile des Betriebssystems.

Befehle können in die Befehlszeile am unteren Rand des Hauptfensters von Siril eingegeben werden. Eine andere Möglichkeit ist es, die Befehle in eine Datei zu schreiben und diese als Skript auszuführen. Um die Datei von der grafischen Benutzeroberfläche aus auszuführen, fügen speichern Sie sie im konfigurierten Skriptverzeichnis oder vernwende Sie in der grafischen Benutzeroberfläche aus das Zeichen `@` in der Befehlszeile wie folgt:

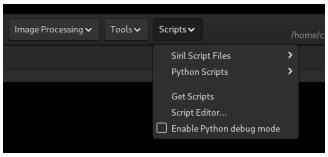
```
@file_name
```

Einige Befehle (*calibrate*, *stack*, und alle Speicherbefehle) können Dateinamen verwenden, die Variablen aus dem FITS-Header enthalten. Das Format des Ausdrucks wird im Detail *hier* erläutert und kann mit dem Befehl *parse* getestet werden.

20.1.2 Benutzung von Skripten

Es gibt vier Wege um ein Skript auszuführen:

- aus der grafischen Benutzeroberfläche, indem Sie das Schlüsselwort `@` in der Befehlszeile verwenden, gefolgt von dem Namen der Skriptdatei im aktuellen Arbeitsverzeichnis,
- aus der grafischen Benutzeroberfläche, über das Menü *Skripte*,

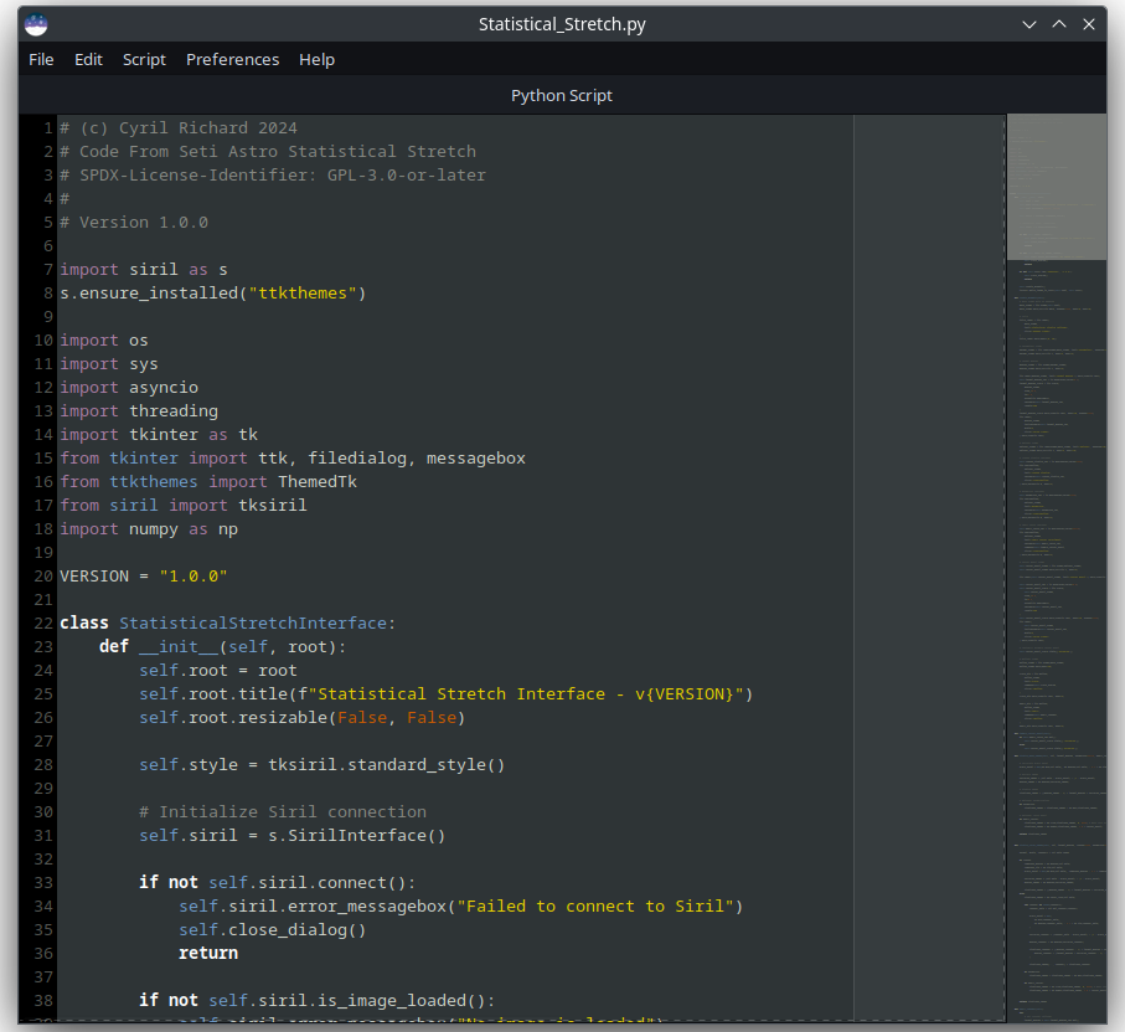


- from the graphical user interface, using the *Script Editor* dialog which can be used to write, save and execute your own scripts or to open and edit an existing script file,
- von der Kommandozeilenschnittstelle (**`siril-cli`** executable), mit dem Argument `-s` gefolgt vom Pfadnamen der Skriptdatei (siehe die Manpage für weitere Informationen).

Wenn ein Skript ausgeführt wird, ist die Benutzeroberfläche mit Ausnahme der Stopp-Schaltfläche nicht interaktiv.

20.1.3 Liste der Skripte aufbauen

Standardmäßig werden bei der Installation von Siril eine Reihe von Skripten automatisch installiert. Diese eingebauten Skripte, die offiziellen Skripte, werden vom Entwicklerteam entwickelt und funktionieren garantiert: Sie sind für bestimmte Standardprozesse gedacht.




```python
# (c) Cyril Richard 2024
# Code From Seti Astro Statistical Stretch
# SPDX-License-Identifier: GPL-3.0-or-later
#
# Version 1.0.0

import siril as s
s.ensure_installed("ttkthemes")

import os
import sys
import asyncio
import threading
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
from ttkthemes import ThemedTk
from siril import tksiril
import numpy as np

VERSION = "1.0.0"

class StatisticalStretchInterface:
    def __init__(self, root):
        self.root = root
        self.root.title(f"Statistical Stretch Interface - v{VERSION}")
        self.root.resizable(False, False)

        self.style = tksiril.standard_style()

        # Initialize Siril connection
        self.siril = s.SirilInterface()

        if not self.siril.connect():
            self.siril.error_messagebox("Failed to connect to Siril")
            self.close_dialog()
            return

        if not self.siril.is_image_loaded():
```

Verzeichnis für Benutzerskripte hinzufügen

Sie können natürlich auch Ihre eigenen <writing-your-own> schreiben und Siril mitteilen, wo sie zu finden sind:

- Klicken Sie auf das Symbol *Hamburger*, dann auf *Einstellungen* (oder drücken Sie `Strg+P`).
- Klicken Sie auf den Abschnitt :*Skripte*.
- Kopieren Sie in eine neue Zeile den Pfad zu dem Ort, an dem sie gespeichert werden sollen (erstellen Sie einen Ordner auf Ihrem Computer oder verweisen Sie auf einen bestehenden).
- Klicken Sie auf das Symbol *Aktualisieren* direkt darunter.
- Klicken Sie auf *Apply*.

Sie können so viele benutzerdefinierte Verzeichnisse nutzen wie sie möchten, indem sie diese einfach der Liste hinzufügen.

Wenn Sie gerade ein neues Skript in einem der Ordner hinzugefügt haben und das Menü aktualisieren möchten, geben Sie den Befehl *reloadscripts* in die Befehlszeile ein oder öffnen Sie den Abschnitt *Einstellungen→ Skripte* und verwenden Sie das Symbol *Aktualisieren*. Dadurch werden alle Ordner der Liste durchsucht und alle Dateien mit der Erweiterung *.ssf gefunden.

> **Warnung**
>
> Es wird dringend empfohlen, Ihre benutzerdefinierten Skripte **nicht** im selben Ordner wie die mit Siril gelieferten Standardskripte zu speichern. Unter Windows können sie bei der Installation einer neueren Version gelöscht werden oder eine korrekte Deinstallation verhindern. Unter MacOS wird dadurch das Gesamtpaket beschädigt und die Verwendung von Siril gänzlich verhindert.
>
> Keine Sorge, die Liste der Skriptspeicherorte ist in Ihrer Konfigurationsdatei gespeichert, so dass Sie sie bei der Installation einer neueren Version wiederfinden sollten.

Skripte aus dem git Repository hinzufügen

Siril supports a git repository at https://gitlab.com/free-astro/siril-scripts This is set to auto update at startup by default, so you will always have access to the latest scripts (auto update can be disabled in preferences). To add scripts from the repository to the Scripts menu, pick the ones you want from the list available in *Preferences → Scripts* or via the *Scripts -> Get Scripts* menu entry. See below for full details of the git repository.

Script Dateien ausführen

Script files can also be run directly from the hard disk using the *Run Script Files...* script menu entry. The filechooser shown by this menu defaults to showing recently used scripts, but you can navigate to choose scripts from anywhere accessible on the filesystem.

Fehlerbehebung

For different reasons, it is possible that the *Scripts* menu is empty. This means that the scripts have not been found. If this is the case, please use the following procedure.

- Klicken Sie auf das Symbol *Burger*, dann auf *Einstellungen*.
- Klicken Sie auf den Abschnitt :*Skripte*.
- Löschen Sie alle Zeilen im Feld *Skript Storage Directories* wie in der folgenden Abbildung gezeigt.
- Wenn Sie das Skript-Repository verwenden, deaktivieren Sie das Kontrollkästchen *Skripte abrufen und aktualisieren...*, markieren Sie es dann erneut und wählen Sie die gewünschten Skripte aus.
- Klicken Sie auf *Apply*.

- Beenden Sie Siril und starten es neu.

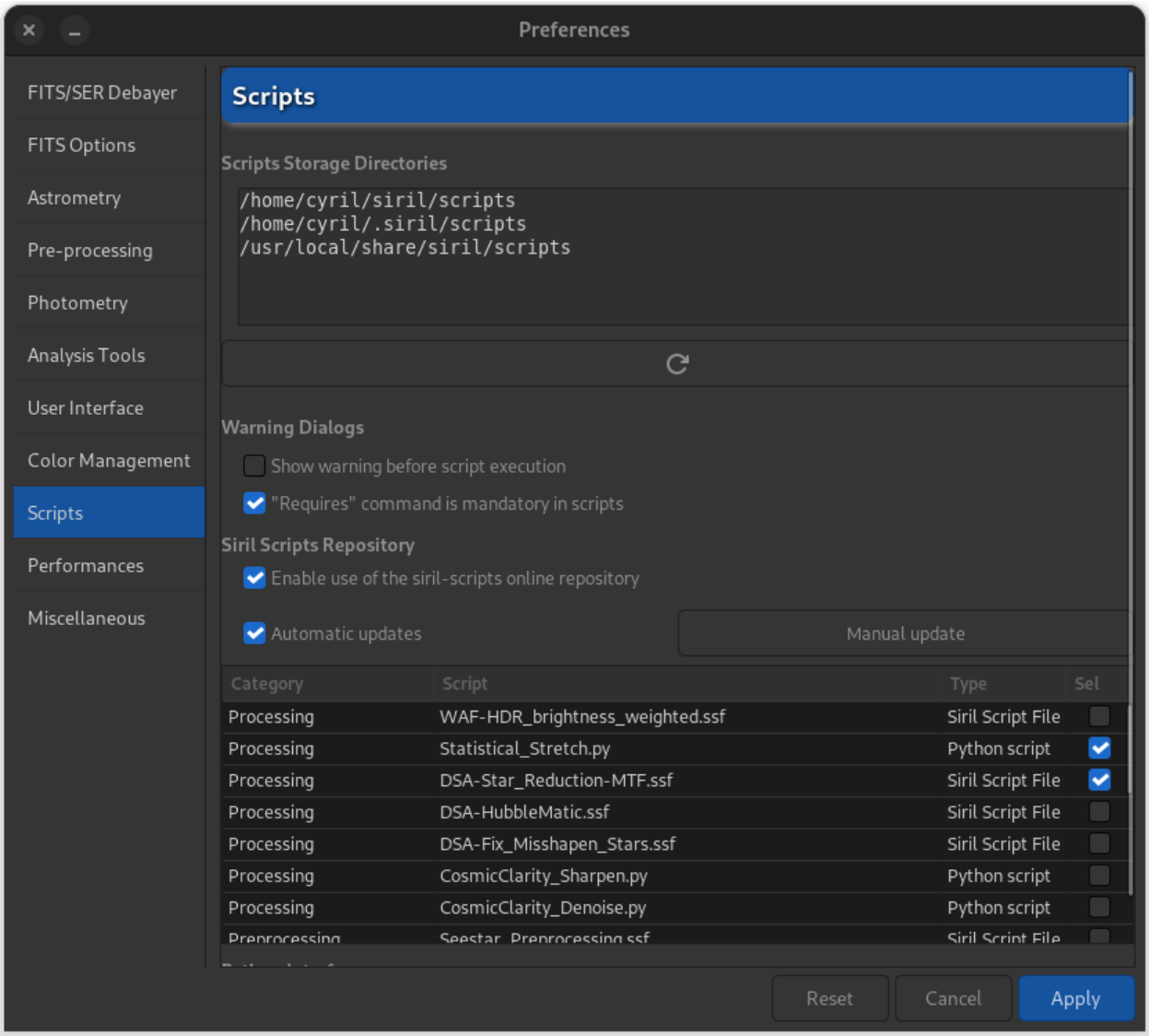
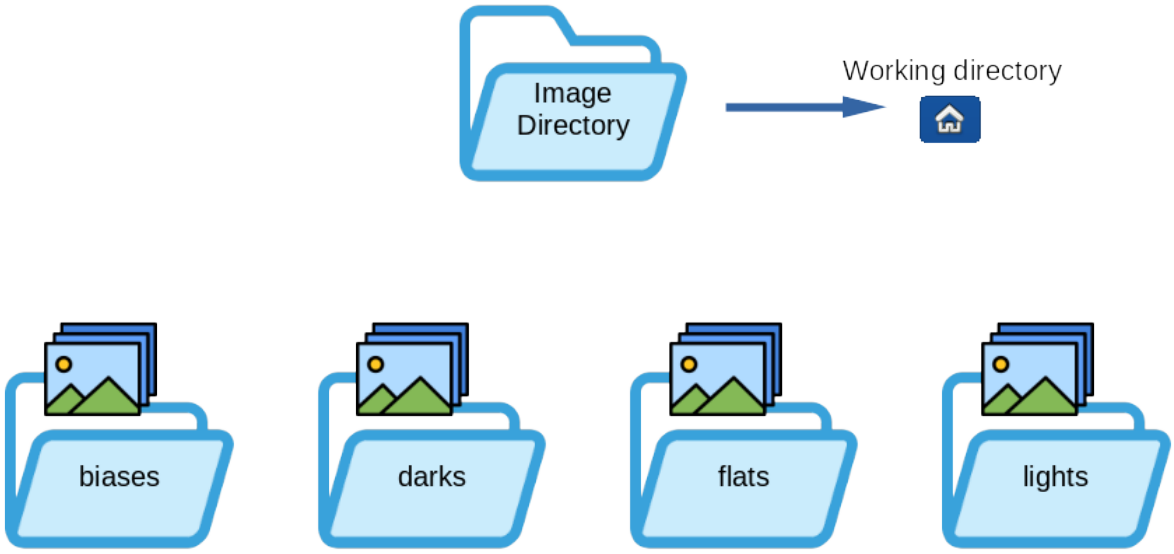


Abb. 2: Skript-Seite der Einstellungen. Die Skripte werden aus den Pfaden geladen, die im *Skript Storage Directories* aufgeführt sind.

20.1.4 Eingebaute Skripte

Alle integrierten Skripte müssen dieser Dateistruktur folgen:

- **Mono_Preprocessing.ssf**: Skript für Kalibrierung von Bildern aus monochromen DSLRs oder Astrokameras, verwendet Bias, Flats und Darks, registriert und stacked die Bilder. Um es zu benutzen legen sie Ihre Dateien (RAW oder FITS) in die Ordner mit den Namen `lights`, `darks`, `flats` und `biases` (im Siril-Standard-Arbeitsordner), dann führen Sie das Skript aus.
- **OSC_Preprocessing.ssf**: Skript für die Kalibrierung von Bildern aus Farb-DSLRs oder Farb-Astrokameras (OSC - One-Show-Color), verwendet Bias, Flats und Darks, registriert und stacked die Bilder. Um es zu benutzen, legen sie Ihre Dateien (RAW oder FITS) in die Ordner mit den Namen `lights`, `darks`, `flats` und `biases` (im Siril-Standard-Arbeitsordner), dann führen Sie das Skript aus.



- **OSC_Preprocessing_BayerDrizzle.ssf**: dasselbe Skript wie oben, aber unter Verwendung von Bayer Drizzle zum ermitteln der Farben. Anwendung: Legen Sie Ihre Dateien (RAW oder FITS) in die Ordner mit den Namen `lights`, `darks`, `flats` und `biases` (im Siril-Standardarbeitsordner) und führen Sie dann das Skript aus. Eine große Datenmenge wird dringend empfohlen, um die Vorteile von Bayer Drizzle zu nutzen und unschöne Artefakte zu vermeiden.
- **OSC_Extract_Ha.ssf**: Skript zur Nutzung mit OSC DSLRs oder OSC Astrokameras, zur Verwendung mit H-Alpha Filtern oder Dual-Band-Filtern. Dieses Skript extrahiert die Ha-Ebene des Farbbildes. Um es zu benutzen, legen sie Ihre Dateien (RAW oder FITS) in die Ordner mit den Namen `lights`, `darks`, `flats` und `biases` (im Siril-Standard-Arbeitsordner), dann führen Sie das Skript aus.
- **OSC_Extract_HaOIII.ssf**: dasselbe Skript wie oben, extrahiert aber die H-Alpha und OIII-Ebenen des Farbbilders. Um es zu benutzen, legen sie Ihre Dateien (RAW oder FITS) in die Ordner mit den Namen `lights`, `darks`, `flats` und `biases` (im Siril-Standard-Arbeitsordner), dann führen Sie das Skript aus. Sie können auch das Menü *Bildbearbeitung*, dann *RGB Zusammensetzung...* verwenden und das Ha-Ergebnis in den Rotkanal und das OIII-Ergebnis in den Grün- und Blaukanal legen, um ein HOO-Bild zu erhalten.

> **Tipp**
>
> Für Besitzer von **SII**- oder **SII-OIII**-Zweibandfiltern gelten die gleichen Skripte. Tatsächlich ist es für einen Farbsensor unmöglich, den Unterschied zwischen **Ha** (656,3 nm) und **SII** (671,6 nm), die beide rot sind, zu erkennen.

- **RGB_Zusammensetzung.ssf**: Dieses in Version 1.2 hinzugefügte Skript registriert monochrome Bilder mit einer globalen Registrierung, beschneidet sie auf ihren gemeinsamen Bereich und nimmt die ersten drei Bilder, um ein Farbbild zu erzeugen. Die Eingabebilder sollten allein in ein Verzeichnis gelegt werden und die Namen `R.fit` (oder mit der konfigurierten Dateierweiterung), `G.fit` und `B.fit` tragen. Das Ergebnis wird `rgb.fit` heißen. Stellen Sie sicher, dass Sie das Verzeichnis `process` zwischen den einzelnen Läufen entfernen.

20.1.5 Skripte bearbeiten

Some .ssf script files contain instructions to edit them, for example the Seestar_Preprocessing script instructs the user that if they find too many images are discarded before stacking, they should "increase the value after -filter-round= in the seqapplyreg command, line 47".

The scripts repository directory is not intended as a user-editable directory: at each update it is forcibly reset to match the state of the remote, so files saved in it are not safe. So how to make such changes?

Find the script in the list in *Preferences->Scripts* and double click it. This will open it in the Script Editor so that you can make the necessary changes and save it in a suitable location (one of the Scripts Storage Directories set at the top of *Preferences->Scripts* is ideal, but it is recommended to give it a different name so you can recognise it in your scripts menu).

> **Tipp**
>
> You can also double click scripts in the list simply in order to examine the code.

Sprache der Skripe

Zu Beginn des Skriptings gab es dank des Beitrags eines Benutzers zwei Versionen der Skripte (Englisch und Französisch). Als Siril 1.2.0 veröffentlicht wurde, wurde beschlossen, nur die englischen Skripte beizubehalten, um die Wartung zu vereinfachen. Wir ermutigen die Benutzer, Übersetzungen der offiziellen Skripte an ihre jeweiligen Communities weiterzugeben, wenn sie dies für notwendig erachten.

20.1.6 Mehr Skripte finden

Es gibt eine ganze Reihe von Skripten, die bei der Installation von Siril nicht mitgeliefert werden. Wir haben jedoch ein Gitlab-Repository für sie eingerichtet. Es steht jedem frei, sich zu registrieren und neue Skripte vorzuschlagen. Wir werden sie entsprechend ihrer Relevanz akzeptieren: Die verwendete Sprache muss Englisch sein.

Siril verfügt über eine Git-Integration, d. h. es kann eine lokale Kopie des Repositorys herunterladen und synchronisieren. Sie können dies aktivieren, indem Sie das Kontrollkästchen *Sirl Skript Online-Repository aktivieren* auf der Registerkarte Scripts des Dialogs Einstellungen aktivieren.

Wenn Sie das Kontrollkästchen aktivieren, werden Skripte aus dem Repository abgerufen und eine Liste der verfügbaren Skripte angezeigt, die entweder als "Vorverarbeitungs-" oder "Verarbeitungsskripte" kategorisiert sind. Da sich unter Umständen eine große Anzahl von Skripten im Repository befinden kann, müssen Sie die Skripte auswählen, die Sie im Menü *Skripte* zur Verfügung haben möchten. Klicken Sie auf das Kontrollkästchen neben dem Namen jedes Skripts, das Sie verwenden möchten, und drücken Sie dann die Schaltfläche *Anwenden*.

Der Inhalt jedes Skripts kann durch einen Doppelklick auf die entsprechende Zeile in der Liste angezeigt werden. Es ist immer nützlich, dies zu tun, um zu prüfen, welche Anforderungen das Skript in Bezug auf vorbereitete Verzeichnisse haben könnte. Wenn Sie ein Skript ändern möchten, können Sie es auch kopieren und in Ihren bevorzugten Texteditor einfügen. (Sie müssen es dann in einem Ihrer benutzerdefinierten Skriptverzeichnisse speichern.)

Wenn das Skript-Repository aktiviert ist, kann Siril das lokale Repository mit dem Online-Repo synchronisieren, entweder manuell oder automatisch. Wenn automatische Aktualisierungen ausgewählt sind, erfolgt die Aktualisierung beim Start von Siril. Eine manuelle Aktualisierung ist auch über die Schaltfläche *Manuelle Aktualisierung* möglich. Dabei werden alle Änderungen aus dem Online-Repository abgerufen und eine Liste der Commit-Nachrichten angezeigt, in denen die Änderungen beschrieben werden. Der Benutzer muss diese bestätigen, um die Aktualisierung durchzuführen.

Wenn Sie die Git-Integration nicht verwenden möchten, können Sie die Skripte auch unter der unten angegebenen Adresse finden und manuell herunterladen. In diesem Fall müssen Sie die heruntergeladenen Skripte manuell in einem Siril bekannten Skriptpfad ablegen. https://gitlab.com/free-astro/siril-scripts.

> **Warnung**
>
> Beachten Sie jedoch, dass diese Skripte nicht unbedingt von den Benutzern, die sie hochgeladen haben, gepflegt werden und möglicherweise nicht auf dem neuesten Stand sind. In diesem Sinne, viel Spaß.

20.1.7 Eigene Skripte schreiben

Eine Skriptdatei ist eine einfache Textdatei mit dem Dateityp *.ssf.

Ein Skript zu schreiben ist nicht schwierig. Es ist eine Folge von Aufrufen von Befehlen, die nacheinander ausgeführt werden. Jeder Befehl muss ausgeführt werden, ohne dass ein Fehler zurückgegeben wird, sonst bricht das Skript ab. Es wird daher dringend empfohlen, die Liste der *Befehle* zu nutzen, um Syntax, Art und Anzahl der verwendeten Parameter zu erlernen. Außerdem sind einige Befehle nicht skriptfähig und werden mit dem Symbol ✕ markiert. Es kann auch nützlich sein, jede Skriptzeile in der Siril-Befehlszeile zu testen. Vielleicht möchten Sie die mitgelieferten Skripte lesen oder Skripte aus dem Repository als Beispiele ansehen (oder sogar ändern).

Jedes Skript sollte einen Kommentar-Header enthalten, der Informationen über das Skript enthält. Ein Beispiel hierfür finden Sie im Folgenden.

```
##############################################
#
# Script for Siril 1.0
# July 2020
# (C) Cyril Richard
# Mono_Preprocessing_WithoutDark v1.0
#
########### PREPROCESSING SCRIPT ###########
#
# Script for mono camera preprocessing
#
# Needs 3 sets of RAW images in the working
# directory, within 4 directories:
#   biases/
#   flats/
#   lights/
#
##############################################
```

Unter der Kommentarüberschrift sollte als erster Befehl *requires* stehen. Dieser Befehl gibt die Mindestversion von Siril an, die für die Verwendung des Skripts erforderlich ist. Zum Beispiel:

```
requires 0.99.4
```

Danach können Sie mti dem eigentlichen Skript beginnen. Befehle stehen in einer eigenen Zeile, und Sie können Ihr Skript mit Zeilen kommentieren, die mit # beginnen.

Each new script created in this way should be placed in a *user-defined folder* for Siril to find them. If you believe your script is of benefit to the wider Siril community you may submit it to the script repository. Instructions on doing so are found in the repository README.

20.2 Pythonskripte

New to Siril 1.4, it is possible to write more advanced scripts using python.

This page is a documentation for script developers. If you want more information about what scripts are available as a user, see *the list of scripts*.

> **Tipp**
>
> Python scripting is still **EXPERIMENTAL** and the API is expected to become more advanced during the 1.5 development cycle.

In its current form, python scripting allows you to:

- Execute any siril command with the added benefit of using a proper programming language so that variables passed to commands can be the result of calculations.
- Access key Siril variables and data structures to aid in scriptwriting.
- Access python extensions.
  - Using the 1.4 scripting interface it is possible to write a plugin that saves the current image, opens it using astropy and applies processing techniques provided by astropy, then saves the result and reopens it as the Siril main image.

- The **tk** python extension can be used to create graphical user interfaces for your scripts so that user input can be obtained directly.

Warnung

Laufzeit Abhängigkeiten

To support installation and configuration of the siril python module and its dependencies, a Python venv ("virtual environment") is configured.

Linux

This requires that there is a functional Python installation on the system _including the `pip` and `venv` modules. These are available by default as part of the flatpak and appimage distributions, but you will need to install them yourself if building Siril from source. On Debian-based systems you need the `python3-pip` and `python3-venv` packages installed; the installation requirements on other systems may vary.

Windows

If you have a Python installation in your system, the script will use that as default. If you don't, don't panic. Siril installed through the official installer also comes with a light-weight Python installation and all the required modules. That is for the official releases. If you build the development version from sources, you will need to have Python installed in your system.

macOS

The necessary dependencies are included in the official MacOS package. We do not provide support for homebrew building, but if you do build yourself on Mac then you will need to ensure you have a functional system python installation.

20.2.1 How does this relate to pySiril?

pySiril is a separate module, published by a separate development team: it is an established product and is already used to implement important helper apps such as Sirilic.

Philosophically, pySiril and the built-in Siril python module are quite different: whereas the built-in module is intended to be called from an already-running Siril to provide scripting functionality and spawns a python3 process to run the script, pySiril is intended for buildin applications that need to call on Siril functionality, and it spawns a Siril process from within its host Python app.

These use cases are quite different, and there is no plan to merge the two modules.

20.2.2 Modul importieren

Siril will take care of setting up a venv (new style python virtual environment) and installing the module in it. In your script you import it using `import sirilpy`.

For convenience, you can abbreviate it as `import sirilpy as s`.

20.2.3 Initialising the Connection

Once you have imported the module, you need to establish the interface with Siril. This is done as follows:

```
import sirilpy as s

siril = s.SirilInterface()
try:
   siril.connect()
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    print("Connected successfully!")
except SirilConnectionError as e:
    print(f"Connection failed: {e}")
```

(The status reporting is optional but is good practice.)

20.2.4 Siril Befehle ausführen

To execute a Siril command, once the interface is established, you use the `siril.cmd()` function. The command and its arguments are provided as a list, for example:

```
siril.cmd("findstar", "-maxstars=1000")
```

will run the `findstar` command, setting a limit of maximum 1000 stars.

The real power of using python to script commands is that you can use python variables in the command using the formatted string notation:

```
x = 1000
siril.cmd("findstar", f"-maxstars={x}")
```

This is a simple example, but the ability to define parameters to pass to image processing functions is a giant leap for Siril scripting compared with legacy Siril script files.

Tipp

Most Siril commands execute an action (`register`, `stack`, `asinh` etc). These will function exactly the same when called from python.

However some simply print information to the Siril log, such as `stat`. Note that these commands do not behave any differently when called from python using `siril.cmd("stat")` or similar: they will still just print information to the Siril log and do not return any useful data to the python script.

In order to access the values from python you need to use a method that gets information about the image or sequence. So in this example, you would call `img = siril.get_image()` and then the statistics would be available in `img.stats`.

Warnung

The `cmd()` method does not have a return value but will raise an exception if the command fails. This provides safe default behaviour - the script will stop on failure of a command and print an exception error message to the Siril console. If you want to handle failure more gracefully, for example by showing an error messagebox, you will need to use an exception handler like this:

```
try:
    siril.cmd("requires", "1.5.0")
except:
    siril.error_messagebox("This script is not compatible with this version of Siril")
    quit()
```

For debug purposes you may wish to have commands return a bool. This can be done using a simple wrapper such as the following:

```
def cmd_with_check(*args):
try:
    siril.cmd(*args)
except Exception as e:
    err_msg = ' '.join(list(args))
    print(f'Exception caught: {e}',)
    print(f"Command failed with arguments: {err_msg} but continuing...")
    return False
return True
```

20.2.5 Access to Siril variables and data structures

The siril module provides access to key Siril variables. The available data is as follows:

- Aktuelles Siril Arbeitsverzeichnis
- Siril user config directory (for storing script-specific config)
- Current Siril image filename
- Aktuelles Siril Bild:
  - Pixeldaten
  - Image ICC profile (as bytes: you will need to use a module such as pillow to convert this raw data to something useful)
  - Image metadata: all the metadata that Siril uses internally, including relevant keywords and image statistics. The full FITS header and a string containing keywords not recognized by Siril are also available.
- Aktuelle Siril Sequenz:
  - Sequenzbild-Pixeldaten
  - Sequence metadata: all the metadata for the currently loaded sequence is available, with the exception of some that relates to details of the sequence format that are abstracted away by the python interface and some relating to photometry series that is not implemented yet. This includes statistics for every channel of every frame, registration data for channels for which it is available of every frame and image quality data for every frame.
  - Sequence frame ICC profiles are not available as these are not generally relevant until after the sequence processing stage. If a convincing use case emerges this could be revisited as an update to the API in the 1.5 development cycle.
  - Sequence frame HISTORY and FITS header strings are not available as they are not considered useful for sequence operations, however the keywords for a sequence frame are available using the `siril.get_seq_frame()` method, which returns a FFit object with the keywords member populated. This provides enough metadata to support any currently identified use case.
- Star modelling data. When stars have been detected in an image the modelling data is available as a list of star parameters for every detected star in the image.

This data is stored in the key data structures `FFit` representing the Siril fits data structure and `Sequence` representing the Siril sequence data structure.

```
# Get the current image
img = siril.get_image()
# Get its dimensions
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
siril.log(f"Current image dimensions: {img.shape[1]} x {img.shape[0]}, {img.shape[2]}␣
↪channels.")
```

Note that the image shape is stored in a typical python order: shape[0] = height, shape[1] = width, shape[2] = channels. The width, height and channels are also directly accessible using the properties `img.width`, `img.height` and `img.channels` once `img = get_image()` has been called.

When the current image is obtained using `siril.get_image()`, the pixel data can be accessed as a numpy array, allowing directly operating on pixel data.

```
import siril
import numpy as np
# Set up the interface as above
# ...

# Try to get the current image, do something to it and update it in Siril
with siril.image_lock():
   img = siril.get_image()
   img.data[:] *= 2
   siril.set_image_pixeldata(img.data)

except Exception as e:
   raise SirilException(f"Error changing pixel data: {e}")
```

We just accessed the pixel array and multiplied every pixel value by 2! Note that we have to call the `siril.set_image_pixeldata()` function to update the data back to Siril. Using `siril.set_image_pixeldata()` you can even update the image to a different size or shape, or change from 16-bit unsigned data to 32-bit floating point or vice versa.

Tipp

In order for `siril.set_image_pixeldata()` to succeed you must claim a lock on the image. This ensures that nothing else can try to update the image at the same time as you are doing, and also ensures that you can't update the image while something else is already doing so. Note that the example above uses the context handler `siril.image_lock()` **before** getting the image from Siril, not just before updating it. This ensures the image is locked throughout the time it is being processed until it is updated back in Siril and the lock is finally released.

In a similar way you can access image statistics, FITS header information, bit depth and so on. You can also access sequence information, for example whether the *n* th image in a sequence is included, what its stats are and so on.

20.2.6 Installation von Python Modulen

You might have noticed we are using NumPy in the previous examples. This is a dependency of the siril module and will be pulled in during initial setup of the venv. Other python modules can be imported using the standard `import` command, for example `import astropy as ap`. However, for ease of ensuring that dependencies are installed if a user doesn't already have them, the module also provides the `ensure_installed()` method. This is used as follows:

```
s.ensure_installed("astropy")
import astropy
```

(or, for another example,

```
s.ensure_installed("tifffile")
import tifffile
```

First a check is done to try importing the module (in this case, astropy or tifffile). If the module is not available for import, an attempt is made to install it. If the install fails, an exception is raised and the script will halt.

> **Tipp**
>
> If a module requires installation this may take a short time, however the delay only occurs on the first run of a script. On subsequent runs the model is already installed, so the `ensure_installed()` check is almost instantaneous. A message in the log notifies the user that module installation is occurring.

If the check succeeds, the module can be imported as normal using `import`.

> **Tipp**
>
> **All other non-core modules except for numpy should be checked with `ensure_installed()` as this automates the installation process for users who don't already have the required modules installed.**

> **Warnung**
>
> All Siril python scripts share the same venv. When importing modules, it is important to avoid overconstraining the version requirements in order to avoid clashes between the modules required by different scripts. Only ">=" version constraints should be used, never "==" or "<=".

```
import sirilpy as s
s.ensure_installed("astropy")
import astropy as ap
```

This will automatically pull in all of astropy's dependencies as well.

20.2.7 Use of External Modules

> **Tipp**
>
> The `siril.FFit` object type is not the same as astropy.io.fits. Whereas astropy.io.fits provides a general file-based interface to all FITS files, siril.FFit provides an interface to the data structure Siril uses to represent FITS images. The two are not directly interchangeable! Either method can be used to obtain the pixel data as a NumPy array.

Whether you obtain the pixel data directly from the `siril.FFit` object or using astropy.io.fits, siril modules such as astropy, photutils and matplotlib provide building blocks for a huge range of image analysis including source detection, image segmentation, source deblending and visualisation, and the ecosystem of python modules provides limitless capability.

```
import sirilpy as s
s.ensure_installed("astropy")
import astropy
from astropy.io import fits
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
with fits.open("filename.fit", mode='update') as image:
   if isinstance(image[0].data, np.ndarray):
      image[0].data *= 2
```

This looks much like our last example - all we do is multiply the pixel data by 2 - but now we have used astropy to open it directly from a FITS file.

20.2.8 API Referenz

Note that the *API* is very new. While we will try very hard to avoid script-breaking changes to the API released with 1.4.0 throughout the stable 1.4 series, some parts of the API may evolve during the 1.5 development cycle.

20.3 Python Script Key Information for Users

20.3.1 Obtaining Scripts

End users should take care to obtain scripts from reputable sources.

- A few fundamentally important scripts are distributed as part of the core Siril package and can be found installed in the system data directory. These are tested to the same degree as the rest of Siril and the development team will treat bug reports on these scripts in exactly the same way as bug reports on any other part of Siril.
- 1.4.0 introduces the Scripts repository. We try to keep a list of the scripts with a short description *here*. Anyone may submit scripts to this repository by making an account on gitlab and submitting a merge request. The Siril team will provide a basic level of scrutiny that the scripts are not attempting to do anything malicious, but we do not accept any responsibility for the correct functionality of scripts written by other people and will not provide support for them - the author should be contacted directly.
- Script authors may choose to distribute scripts independently, as has been done in the past. This is fine: scripts can be downloaded and added to a script directory or run via the script editor from anywhere on the filesystem. In this case the Siril team have nothing at all to do with the scripts and cannot even assure you that they will do no harm, so ensure you trust the author.
- Siril does not support automatic verification of signed scripts. This is because script signing can provide a false sense o security ("the script is signed so it can't do anything bad"). This is false: all a script signature says is that the script was signed by a particular person and hasn't been changed since. In other words, it verifies the origin of a script, but says nothing about whether it functions correctly or is safe. **A signed and verified script could still destroy your data if that is what it is written to do!** In the current implementation of python scripting, the majority of scripts are distributed through the centralised script repository in which case the origin is known and scripts have been through a basic level of scrutiny by the development team prior to being merged. Some scripts may be distributed through other channels and downloaded manually, and there is nothing preventing an author providing a sha256sum or a .gpg signature for scripts distributed in this way - you will just need to verify them manually after downloading.

20.3.2 Troubleshooting Common Issues

Siril's python script system is quite complex: a lot goes on under the hood, and this is all coordinated by clever use of a Python venv ("virtual environment") which ensures that the python used by Siril is separated from the system Python installation and potentially from other venvs that may be in use by other software. However there are a few things to be aware of:

- venvs are tied to a specific version of python. This means that if your system updates from (e.g.) Python-3.12 to Python-3.13 the venv will no longer point to the right Python program or the right bytecode libraries for compiled modules. **When you upgrade to a new version of Python you must reset the Siril venv**.

- script dependencies are controlled using pip and the pypi archive. This is very good but not 100% foolproof, and in rare cases you may find a dependency issue has arisen that causes a problem of some kind. **If you find an unexplained problem with a script that others aren't experiencing, resetting the venv may solve it.** (And if you find a combination of scripts that can repeatably cause a dependency issue like this, please report it as an issue on the siril-scripts gitlab site as it will help us to improve advice to script authors.)

Resetting the venv is easy: click the *Get Scripts* menu item in the Scripts menu and scroll down to the bottom. Here you will find a "Reset venv" button: clicking it will reset the venv and create a new one. After doing this, the next time you run a script it will need to reinstall its dependencies so the first startup will be a bit slower than ususal (depending on network speed and whether any of the required modules are locally cached).

20.4 Python Script Key Information for Authors

We welcome all script authors who wish to write scripts for Siril. However supporting Siril, while fun, takes a lot of time and we have neither the time nor the desire to support other people's work as well as the core Siril software. There are therefore some guidelines to ensure that the division of responsibility is clear.

- All scripts should feature the author's name or handle, as well as a means of contact (website, YouTube channel, forum etc.) where they can be reached with comments or bug reports. For authors publishing independently this is of course guidance, however for authors wishing to publish their scripts via the siril-scripts repository this is mandatory. Scripts will not be accepted unless these details are included.
- Only open-source scripts will be accepted in the Siril scripts repository. Closed source scripts (including ones that feature a source code wrapper that calls a closed source program) must be independently published.
- Authors should also follow the script programming guidelines on the API page. These guidelines ensure that scripts can be used by end-users with minimal friction.

20.4.1 Siril-Scripts Repository Code Standards

As a script author you can write whatever scripts you like for your own use. However for scripts to be published to the siril-scripts repository we need to require some coding standards in order to ensure that they work with the standard Siril packages on each of the three supported operating systems. The requirements are:

- Scripts should work with versions of Python >= 3.9. This version matches the oldest Python version in our binary packages, and will be adjusted over time to match the packages.
- Scripts should be single-file scripts. The repository was originally designed for legacy .SSF scripts and the assumption of one file per script remains. We understand that it can be good for code readability to develop complex python programs by splitting them into multiple files. This is fine when working locally, however the multiple files would need to be combined into a single file for submission to the scripts repository.
- If you want to implement a GUI in your script you should use the pyQt6 toolkit.
  - pyQt6 uses the well known Qt framework and it has now been verified as installing correctly from pypi on all the supported platforms. Qt6 is much faster than Tkinter for scripts that themselves have to do significant graphics manipulation, such as pan-and-zoom previews, and it has a nice GUI designer application available in Qt Creator.

> **Warnung**
>
> tkinter is now deprecated: new Siril scripts should not use it and should be written using pyQt6 instead. Although tkinter is a core Python module it is outdated, very slow and has issues with Wayland desktops on Linux. tkinter scripts will continue to work throughout the 1.4 stable series, but there is a considerable amount of added complexity in the sirilpy code to support tkinter. Since pyQt6 provides a much better alternative has now been confirmed to work on all the platforms Siril supports, the tkinter support code (the tksiril and

tkfilebrowser submodules) will be removed during the 1.5 development cycle and will be removed before 1.6.0.

Warnung

Scripts using GTK GUIs will not be accepted into the scripts repository. This GUI framework requires substantial system dependencies. Although Siril itself is coded in GTK, unfortunately the python gi packages required to use it in python scripts are broken on Windows and we are unable to bundle them. If you wish, you can install the dependencies yourself and write scripts using GTK and they may work fine on your own machine, but this is not supported. We will not provide help in getting the prebuilt packages to work with unsupported packages; if you are determined to try this, you will likely need to build Siril from source.

- Avoid import version specifications that specify a package version using "<", "<=" or "==". These constraints will result in conflicts between scripts, which all share the same venv. Specifying a package version using ">=" is fine. Remember to use `sirilpy.ensure_installed()` before importing non-core python module dependencies, to automate the process of installing them.
- Avoid importing packages that require system packages (i.e. ones that cannot be installed using `python3 -m pip install`)
- Ideally, check that your script imports will install on all three target operating systems. If not then the development team will check them after submission, and you will need to fix any issues that occur.

20.4.2 Known Good Modules

A list of "known good" packages that we have checked will import without problem, and with all their dependencies, on all the target operating systems, is provided below:

- astropy
- astropy-healpix
- Astroquery
- ccdproc
- GaiaXPy
- matplotlib
- numpy
- opencv-python
- pandas
- photutils
- pillow
- pyfftw
- pygaia
- pyQt6
- scipy
- tk
- ttkthemes

> **Tipp**
>
> For machine learning / AI scripts, onnxruntime is very strongly recommended. It is cross-platform, it is hardware-agnostic and it falls back gracefully to CPU support if a GPU or GPU system libraries are unavailable. The `sirilpy.utility` submodule provides an `ONNXHelper` class to ensure the correct hardware- and OS-specific version of onnxruntime is installed.

20.4.3 Machine Learning Modules

A variety of machine learning modules exist which can be used to write "AI" scripts. However the state of support for heterogenous computing (i.e. GPU use) is patchy and often requires system libraries such as CUDA or ROCm to be installed. The following modules are regarded as key for GPU acceleration and either are supported with helper classes or have helper classes in development.

- **onnxruntime** - used in GraXpert and Riccardo's AberrationRemover.py script, this package is available for all the target OSes. There are different backends to support different GPUs, but in some cases these are fragile. A helper class `ONNXHelper()` is available to help with identification and installation of the optimum package as well as ensuring correct runtime fallback to CPU. Note that currently on Windows the ONNXHelper will select the DirectML runtime for all supported GPUs. Even though in theory the CUDA-based onnxruntime-gpu package might be a bit faster, it unfortunately depends heavily on system libraries being installed and configured correctly and the installation is considered too fragile to be reliable in the kind of automated python environment used by Siril. If users wish to use it and are confident they can get it to work on their system they can install it manually using `sirilpy.ensure_installed("onnxruntime-gpu")` but this is not supported and related issues will be closed.
- **torch** - This is a heavyweight among ML frameworks and where GPU support was once very much CUDA-based it is now broadening its coverage: support for NVidia and Apple Silicon GPUs is good, and support for AMD GPUs (using ROCm) and Intel GPUs is in development (ROCm is supposedly stable enough to use on Linux, not yet on Windows). Unfortunately torch has some dependency issues (it is excessively strict about the version of CuDNN it requires) which causes problems using it in the same venv as some other modules (including jax - see below) that pull in the nvidia-cudnn package. A `TorchHelper()` class is available which manages this by first installing torch, then reinstalling it with the --no-deps flag, which prevents it complaining if another package updates the dependencies to a higher version number. This is really ugly and it's unfortunate that it's necessary, but it works. If you envisage using CUDA versions of both torch and jax, it is better to install torch first and then jax.
- **jax** - Not a dedicated ML framework as such but a means of offloading computation to the GPU. jax provides a mostly numpy-compatible module called jax.numpy which can greatly accelerate many mathematical array processing operations by offloading them to the GPU and also provides functions for gradient computation that numpy does not. Bearing in mind that the slow part of GPU processing is the data transfer between system memory and GPU memory, jax provides a jit compiler to precompile its offloaded workloads but it still requires careful algorithm design to keep data on the GPU for as long as possible, only returning the final result. Jax has good GPU support on Linux but only CPU support on Apple and Windows, although experimental Metal support and Windows GPU support is in progress. This is therefore not recommended for actual scripts at present, but is highlighted as an interesting framework to develop against in the (hopefully) near future when its support broadens. A `JaxHelper()` class is under development that will help with selection of the correct jax package to install and testing to confirm it works properly.
- **Dependencies** Note that both torch and jax use GPU library pypi packages, so installation of them may pull in very large chains of dependencies - the full set of NVIDIA packages (CUDA, CuDNN etc.) is around 1GB. So when using these in scripts be sure to warn users that they may incur a large and potentially slow download on first use!

20.4.4 Excluded Modules

Completely avoid using the following modules, as they are known to cause problems on one or more OSes:

- healpy (doesn't work on Windows; astropy_healpix can be used instead)
-

20.4.5 Modules with System Dependencies

The following modules require installation of system packages which cannot be automated using pip, and therefore should not be directly used in Siril scripts:

- `gi` (pygtk etc - requires system package installation)
-

> **Warnung**
>
> **We cannot emphasise enough that such packages are not supported.** If they work for you locally, that's a lucky bonus for you. They generally won't work with any of the prebuilt binary packages (AppImage, flatpak, Windows, MacOS). These kind of packages are specifically not supported and we will not provide help with getting them to work. All tickets relating to this kind of question will be closed as a duplicate of #1527.
>
> Yes, we know it's a shame. In fact we would have liked to use GTK as the toolkit for writing sirilpy scripts, but unfortunately the binary wheel has been broken on Windows for a long time and doesn't show any sign of being fixed. We can't release core functionality that doesn't work on one of the target OSes.

20.4.6 Workaround for Modules with System Dependencies

It may still be possible to write projects that use these kind of packages, but you will need to package them with their dependencies and distribute them independently of Siril or the scripts repository, and the sirilpy module can still be used to achieve integration with Siril. If you want to, you can provide a wrapper script that can be distributed in the scripts repository and just initializes your main program. We will not help with any of that. As stated above, use of these packages in Siril scripts is not supported.

20.4.7 Closed Source Scripts

The python interface will happily run precompiled .pyc files. Siril is a Free and Open-Source software, and we therefore do not encourage development of closed-source scripts, however we don't forbid it either. Practically we can't, because even stringent interpretations of the GNU Public License allow for writing an open source shim that sits between Siril and a closed source application. Moreover we are in favour of freedom, and while we choose to release Siril under the GPL to provide freedom to our users, we also respect the freedom of developers to choose how they release their own work.

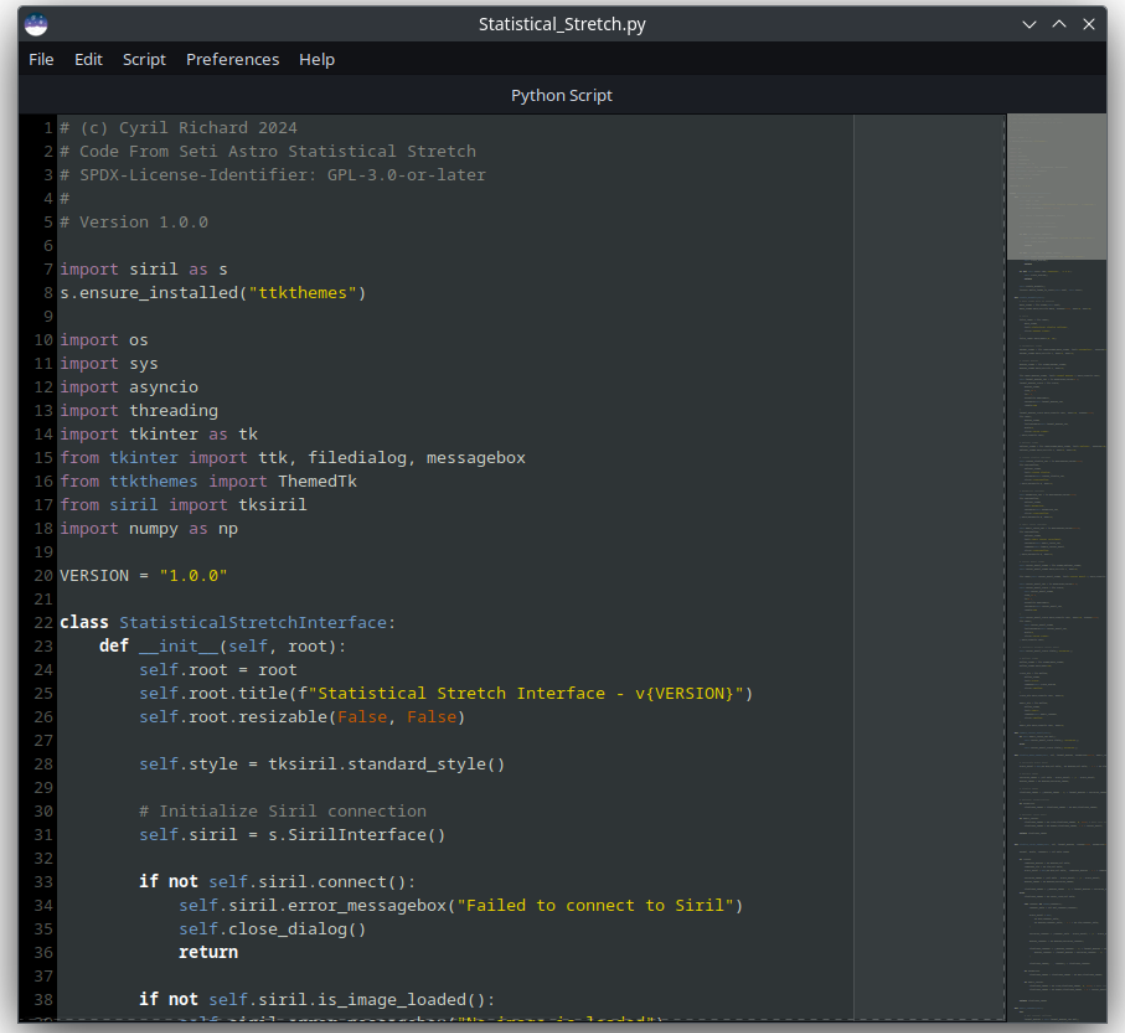
There is some history here, too: Siril has since 1.2.0 provided a built-in interface to Starnet, however Starnet was originally open source and only closed its doors later on.

However, if an author chooses to release a Siril script as a closed-source .pyc then they need to arrange all matters to do with distributing it: as we are unable to inspect the code ourselves to perform even the most cursory checks, we cannot host it in the Siril-scripts git repository. And of course, responsibility for supporting such products is entirely a matter for the author - the Siril team will offer no advice on such products.

Note that scripts must be compiled separately for each version of python: helloworld.py would be compiled with `python3 -m compileall helloworld.py` which would generate helloworld.cpython-<version>.pyc, where <version> would be 312 for Python 3.12, 313 for Python 3.13 and so on depending on the version of python used to compile the script. This is an inherent inconvenience to closed source scripts.

20.5 Skripteditor

Siril contains a script editor, accessible via the Scripts menu. This provides a feature rich code editor tailored for writing both Siril Script Files and Python scripts. The theme (light / dark) of the script editor follows that set in the main *Preferences* dialog. The main editor window is shown below:



For most of the functionality shortcuts exist, which are as standardised as possible.

20.5.1 Dateimenü

The file menu provides the usual functionality:

- **New**: begins a new file. If there is unsaved text in the buffer a confirmation dialog will be presented to give the user a chance to save their previous work.
- **Open**: open a file from disk. If If there is unsaved text in the buffer a confirmation dialog will be presented to give the user a chance to save their previous work.

- **Recent files**: provides quick access to open recent files.
- **Save**: saves the current file with its existing filename. If the file is not yet saved then this behaves in the same way as Save as, prompting for a filename.
- **Save as**: saves the current file, prompting for the filename to save it with.
- **Close**: closes the file and the script editor window. If there is unsaved text in the buffer a dialog will be presented to give the user a chance to save their work.

Unsaved files are indicated by an asterisk (*) next to the filename in the titlebar. When changes are saved, the asterisk will disappear.

20.5.2 Edit Menu

Das Bearbeiten Menü stellt folgende Funktioen bereit:

- **Undo**: undoes the most recent change in the script. The script editor has an unlimited undo / redo buffer.
- **Redo**: Wiederholt die letzte Rückgängig-Aktion im Skript.
- **Cut**: cuts the current selection to the clipboard. If no selection is made, the entire current line is cut.
- **Copy**: copies the current selection to the clipboard.
- **Paste**: pastes the current clipboard content to the cursor position.
- **Find**: Presents the *Find overlay* which provides an active search capability. The editor will scroll to the first occurrence of the search term as you type. All occurrences are highlighted in the editor view and in the minimap, if enabled.

20.5.3 Skriptmenü

The script menu provides the following functionality:

- **Run**: runs the current script. Note that this obeys the current script type selection (Siril Script File / Python Script) and this must be correct for the script you are editing. The script type is automatically set based on the file extension when loading a file, but you must set it here if you have not yet saved your script. At startup it defaults to Python Script. Note that owing to a shortcut clash in some desktop environments, as well as the listed `F5`, `Ctrl-R` also works to run the current script.
- **Python scripts**: sets the current script mode to Python scripts. This affects the syntax highlighting as well as the manner in which the script is run.
- **Siril Script Files**: sets the current script mode to Siril Script Files. This affects the syntax highlighting as well as the manner in which the script is run.
- **Enable test arguments**: Shows an entry below the main script view. This can be used to enter arguments that will be passed to the script when run from the editor. Python scripts can take arguments to set parameters: this provides an alternative to using a GUI to set parameters. Even if a GUI is provided in a script, the ability to pass arguments means that it can be used as a Siril command via the `pyscript` command, and can therefore itself be used from within scripts.
- **Enable python debug mode**: This toggle changes the behaviour of the sirilpy module to support debugging. When the SirilInterface is created at the start of the script, a modal information box will appear showing the process ID of the python interpreter, which can be used to connect a debugger. Also, all interface timeouts will be set to None, to allow for prolonged inspection of the state of a running script.

  Note that the test arguments have no effect on Siril Script File scripts, as these have no ability to handle arguments.

20.5.4 Voreinstellungen

Das Menü „Einstellungen“ bietet die folgenden Optionen:

- **Highlight syntax**: when enabled, Siril will use language-aware syntax highlighting to highlight aspects of your script.
- **Enable right margin indicator**: when enabled, Siril will show a right margin indicator at a set column number. This is useful for helping to avoid excessively long lines of code and maintain readability.
- **Right margin position**: this will present a small dialog allowing the user to set the column number of the right margin indicator discussed above. This defaults to 80.
- **Enable bracket matching**: when enabled, if the cursor is on a bracket the editor will highlight the matching bracket. This can be useful to avoid bracket mismatches.
- **Show line numbers**: when enabled, line numbers are shown to the left of the editor view. This can help with debugging and navigating code.
- **Show line markers**: when enabled, line markers are shown to the left of the editor view. No functionality currently uses line marks, but the ability to view them is provided to support future developments.
- **Highlight current line**: when enabled, the current line is highlighted in the editor view.
- **Enable auto-indentation**: when enabled, on pressing Enter the new line will begin at the same level of indentation as the previous line.
- **Indent on tab**: when enabled, pressing tab with a selection made that covers multiple lines will cause all the selected lines to be indented a level, and pressing shift-tab will cause them to be unindented a level.
- **Enable smart backspace**: when enabled, pressing backspace with whitespace to the left of the cursor will delete whitespace back to the previous indentation level.
- **Smart Home / End**: when enabled, Home and End will move the cursor to the first or last non-whitespace character of the line respectively, rather than the absolute first or last character of the line.
- **Show spaces and tabs**: when selected, spaces and tabs will be shown with visible characters (central dots for spaces, right-facing arrows for tab characters).
- **Show newlines**: when selected, newlines will be shown with visible characters (down-and-left bent arrows).
- **Show minimap**: when selected, a minimap will be shown to the side of the editor view, supporting navigation and location of `Find` occurrences.

20.5.5 Help

The help menu provides API help for both Siril Script Files and Python scripts.

- **Python API Reference**: this opens the Python API page of the online manual in the default browser.
- **Commands Reference**: this opens the Commands Reference page of the online manual in the default browser, which is essentially the API for Siril Script Files.

20.6 Python Skriptbeispiele

This section highlights examples of python scripting. These are simple examples intended to provide an introduction to the API, and are explained with commentary.

For a more complete example of a script using a TKinter interface to provide a full-featured interface to an external program, see the CosmicClarity_denoise.py and CosmicClarity_sharpen.py scripts which are available in the Siril scripts repository.

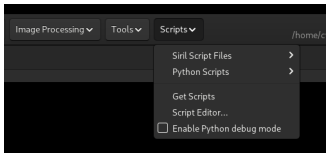


Abb. 3: Skriptmenü

20.6.1 Hello, Siril!

It is traditional to start with a Hello, World! example, so here is "Hello, Siril!"

```
import sirilpy as s
from sirilpy import SirilConnectionError

siril = s.SirilInterface()

try:
   siril.connect()
   siril.log("Hello, Siril!")

except SirilConnectionError as e:
   print(f"An error occurred: {e}")
```

Das war's. Alle Siril-Python-Skripte sollten das Siril-Modul importieren: Dieses wird automatisch von Siril bereitgestellt und muss nicht installiert werden.

It allows connection to Siril through the **SirilInterface** class and gives access to Siril methods: in this case the `log()` function does a tiny bit of adjustment (adding a `\n` newline character) and calls the internal function `siril_log_message()` to display the result in Siril's log tab. Note that `log()` can take an optional argument specifying the color: if you wanted "Hello, Siril!" to print in green you would change the line to `siril.log("Hello, Siril!", s.LogColor.GREEN)`.

Note that there is generally no need to call `siril.disconnect()` call, as it is not really necessary: at the end of a script the method will be called automatically in an onexit handler.

Also note that although you can initialize more than one `SirilInterface()` object, a script can only connect via one at a time (though multiple scripts can be connected to Siril simultaneously). There is generally no need to create more than one `SirilInterface()`.

20.6.2 Hello, astropy!

This script shows how to save a temporary working copy, open it using astropy.io.fits and modify it before saving the file and reopening it in Siril.

```
import sirilpy as s
import tempfile
import os
import gc
import numpy as np # numpy is a dependency of the siril module and
# will always be available, so no need to use s.ensure_installed() here

s.ensure_installed("astropy")
import astropy
from astropy.io import fits
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
siril = s.SirilInterface()
try:
   siril.connect():
   print("Connected successfully!")
except SirilConnectionError as e:
   print(f"Connection failed: {e}")
   quit()

def hello_astropy():
   siril.log("Starting the process...")

   temp_filename = None
   try:
      # Create a temporary file
      with tempfile.NamedTemporaryFile(suffix=".fits", delete=False) as temp_file:
            temp_filename = temp_file.name
            siril.log(f"Temporary file created: {temp_filename}")

      # Save current file
      siril.cmd("save", temp_filename)
      siril.log(f"FITS file saved: {temp_filename}")

      # Open and modify FITS file using Astropy
      with fits.open(temp_filename, mode='update') as image:
            siril.log("Opened FITS file with Astropy")

            # Modify the FITS file data
            if isinstance(image[0].data, np.ndarray):
               image[0].data *= 2
               siril.log("Modified FITS file data using Astropy (multiplied pixel values␣
↪by 2)")

            image[0].header['COMMENT'] = "Modified by Astropy"
            siril.log("Added COMMENT to FITS file header")

            image.flush()  # Ensure changes are written
            siril.log(f"Changes saved to FITS file: {temp_filename}")

      # Load back into Siril
      siril.cmd("load", temp_filename)
      siril.log(f"FITS file loaded: {temp_filename}")

      siril.log("Process completed successfully")

   except Exception as e:
      siril.log(f"An error occurred: {str(e)}")
      raise  # Optionally re-raise if needed for further handling

   finally:
      # Clean up: delete the temporary file
      if temp_filename and os.path.exists(temp_filename):
            try:
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                os.remove(temp_filename)
                siril.log(f"Temporary file deleted: {temp_filename}")
            except OSError as e:
                siril.log(f"Failed to delete temporary file: {str(e)}")

        # Garbage collection using gc.collect() is not required at the end
        # of a script as it gets run automatically just after the script
        # hands back to the interpreter

# Run the function
if __name__ == "__main__":
    hello_astropy()
```

Note the use of error handling in this example, using the try: and except: blocks, and the cleanup in the finally: block.

20.6.3 Pixel Data and Metadata

The sirilpy python module offers direct access to a wide range of data.

```
import sirilpy as s
import numpy as np
siril = s.SirilInterface()
try:
    siril.connect():
    print("Connected successfully!")
except SirilConnectionError as e:
    print(f"Connection failed: {e}")
    quit()

def siril_module_example():
    try:
        # Lock the main image to prevent anything else altering it while we
        # are working on it
        with siril.image_lock():
            # Get the global fits object
            fit = siril.get_image()

            siril.log(f"The value of the first pixel is: {fit.data.flat[0]}")
            fit.data[:] *= 2
            siril.set_image_pixeldata(fit.data)

        siril.log(f"The value of the first pixel is now: {fit.data.flat[0]}")
        siril.log(f"Data type: {fit.data.dtype}")
        siril.log(f"Array shape: {fit.data.shape}")
        history = fit.history
        siril.log(f"History: {history}")

    except AttributeError as e:
        siril.log(f"Error: {e}")
        fit = None  # Ensure the FFit object is released

# Run the function
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
if __name__ == "__main__":
    siril_module_example()
```

As before, we import the siril module: we now also import numpy, which we will use to operate on the pixel data in the image loaded in Siril. Note that there is no need to use the `import_or_install()` method, as numpy is a dependency of the siril python module and will have been preinstalled in the venv.

Next, we demonstrate various ways of interacting with the image data and metadata:

```
# Get the global fits object
with siril.image_lock():
    fit = siril.get_image(True) # True indicates that we want the pixel data
    # not just the metadata. It is not required as this is the default, but
    # if you only want the metadata you can specify False here to save a bit
    # of memory.

    siril.log(f"The value of the first pixel is: {fit.data.flat[0]}")
    fit.data[:] *= 2
    siril.set_image_pixeldata(fit.data)
    siril.log(f"The value of the first pixel is now: {fit.data.flat[0]}")
```

We get the current image using `siril.get_image()`

Tipp

Note that `siril.fits` is not at all the same as `astropy.io.fits`. While astropy.io.fits provides a general-purpose interface to all FITS files, siril.fits provides an interface matching the fits data structure used in Siril.

We then access the pixel data as a NumPy array and carry out operations on this array to modify the image data. In this case we multiply all the pixel values by 2.

After we have finished modifying the image, we call `siril.set_image_pixeldata(fit.data)` to update the pixel data in Siril and trigger the Siril GUI to redraw to show the updated image. Along the way we use `siril.log()` to highlight the change to the first pixel.

Next we show the array datatype and shape:

```
siril.log(f"Data type: {fit.data.dtype}")
siril.log(f"Array shape: {fit.data.shape}")
```

Next we show how we can access the image history. This is provided as a list of individual HISTORY FITS headers, so we join them together and add newline characters before logging.

```
history = fit.history
siril.log(f"History: {history}")
```

Full details of the API can be obtained by importing the Siril module and calling the `help(sirilpy)` command, or more specifically by calling `help()` for the particular part of the API you want to know about, e.g. `help(sirilpy.FFit)`.

As docstrings are not translatable, for translations of the API help you will need to consult the API documentation page in this online manual.

20.6.4 Siril.plot Submodul

This script shows how to use the siril.plot submodule to create a simple plot with three series. The first is a simple x-y plot with lines; the second is an x-y plot with error bars; and the third is another simple x-y plot but shows use of numpy arrays as well as how a SeriesData object can be created and added directly to the PlotData object using `add_series_obj`.

```
import sirilpy as s
import numpy as np
siril = s.SirilInterface()
try:
   siril.connect()
except SirilConnectionError as e:
   print(f"Error: failed to connect to Siril: {e}")
   quit()

# Series data using lists of floats
plot_data = s.PlotData(
   "Example Plot",
   "X Axis",
   "Y Axis",
   "example_plot_lists",
   True,
   datamin = [0.0, 0.0]
   )

plot_data.add_series([1.0, 2.0, 3.0],
                     [4.0, 5.0, 6.0],
                     "Series 1 (Lists)")

plot_data.add_series([4.0, 5.0, 6.0],
                     [7.0, 8.0, 9.0],
                     "Series 2 (Lists with errors)",
                     s.PlotType.MARKS,
                     [0.35, 0.4, 0.45],
                     [0.35, 0.4, 0.45])

x_arr = np.array([1.0, 2.0, 3.0, 4.0], dtype=np.float32)
y_arr = np.array([1.0, 4.0, 9.0, 16.0], dtype=np.float32)
series3 = s.SeriesData(x_arr, y_arr, "Series 3 (np.arrays)")
plot_data.add_series_obj(series3)

# Serialize using lists
siril.xy_plot(plot_data)
```

20.6.5 Vorlagen für Skripte mit GUIs

The following templates show how to create scripts with a GUI. The first template is the simplest and only provides a GUI interface:

Quellcode 1: gui_template.py

```
# Template for Siril script with GUI only (no CLI)
# SPDX-License-Identifier: # insert license identifier here
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
# insert author's name and details here

"""
This script template provides an exemplar for use in constructing sirilpy
scripts that provide a Tkinter GUI. Such scripts are only usable via the GUI.
This is a simplified version without command-line argument support.
"""

# Core module imports
import os
import sys
import math
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import numpy as np

import sirilpy as s
from sirilpy import tksiril, SirilError
s.ensure_installed("ttkthemes")
# Add other calls to ensure_installed() here to ensure that any non-core
# modules that are required by the script are installed and available
# to be imported.

# Note: pylint will complain that the following imports are not at the
# top of the module. This deviation from python style is required in order
# to run ensure_installed() to ensure that any non-core modules are
# available before attempting to import them.
from ttkthemes import ThemedTk
# Add any additional imports here

VERSION = "1.0.0"

def template_algorithm(fit,
                       example_float_var,
                       example_bool_var,
                       example_file_path_var):
    """
    Template processing function, insert your algorithm here.
    This is the non-GUI part of the script and is entirely up to you to write.
    Although presented in the template as a single method it may comprise
    classes, methods and anything else.
    """
    print(f"Argument: {example_float_var}")
    print(f"Bool var: {example_bool_var}")
    print(f"File path: {example_file_path_var}")
    return fit

class TemplateScriptInterface:
    """ This class provides the GUI and related callbacks """
    def __init__(self, root):
        self.root = root
        self.root.title(f"Template Script Interface - v{VERSION}")
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        self.root.resizable(False, False)
        self.style = tksiril.standard_style()

        # Initialize Siril connection
        self.siril = s.SirilInterface()

        try:
            self.siril.connect()
        except s.SirilConnectionError:
            self.siril.error_messagebox("Failed to connect to Siril")
            return

        # Initial checks: example - check if an image is loaded
        if not self.siril.is_image_loaded():
            self.siril.error_messagebox("No image is loaded")
            return

        # Check if the version of Siril is high enough
        try:
            self.siril.cmd("requires", "1.3.6")
        except s.CommandError:
            return

        # Create the UI and match its theme to Siril
        self.create_widgets()
        tksiril.match_theme_to_siril(self.root, self.siril)

    # This function is used to round values to n decimal places, it is
    # always useful if you have sliders in your UI.
    def _floor_value(self, value, decimals=2):
        """Floor a value to the specified number of decimal places"""
        factor = 10 ** decimals
        return math.floor(value * factor) / factor

    # Here is an example function to round the displayed value of a
    # variable to 2 decimal places. You will need one of these for each
    # slider control you have.
    def _update_example_float_variable_display(self, *args): # pylint: disable=unused-
↪argument
        """Update the displayed template variable value with floor rounding"""
        value = self.example_float_var.get()
        rounded_value = self._floor_value(value)
        self.example_float_var_display_var.set(f"{rounded_value:.2f}")

    def _browse_file(self):
        """
        Use a TK filedialog to browse for a file. Note that this callback is
        specific to the variable being updated; you might need to have several
        similar callbacks if you have more than one file selection widget.
        """
        filename = filedialog.askopenfilename(
            title="Select File",
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
            initialdir=os.path.expanduser("~")
        )
        if filename:
            self.example_file_path_var.set(filename)

    def create_widgets(self):
        """Create the GUI's widgets, connect signals etc. """
        # Main frame with no padding
        main_frame = ttk.Frame(self.root)
        main_frame.pack(fill=tk.BOTH, expand=True, padx=0, pady=0)

        # Title
        title_label = ttk.Label(
            main_frame,
            text="Template Script",
            style="Header.TLabel"
        )
        title_label.pack(pady=(0, 20))

        # Parameters frame
        params_frame = ttk.LabelFrame(main_frame, text="Parameters", padding=10)
        params_frame.pack(fill=tk.X, padx=5, pady=5)

        ##############################################################
        # Example control with an embedded variable in its own frame
        ##############################################################

        template_variable_frame = ttk.Frame(params_frame)
        template_variable_frame.pack(fill=tk.X, pady=5)

        ttk.Label(template_variable_frame,
                  text="Template example_float_var:").pack(side=tk.LEFT)

        # Initialize variables with default values
        self.example_float_var = tk.DoubleVar(self.root, value=1.0)
        self.example_float_var_display_var = tk.StringVar(self.root, value="1.00")

        # Add trace to update display when slider changes
        self.example_float_var.trace_add("write",
                                         self._update_example_float_variable_display)

        example_float_var_scale = ttk.Scale(
            template_variable_frame,
            from_=0.0, # set your range minimum here
            to=1.0,    # set your range maximum here
            orient=tk.HORIZONTAL, # oriented horizontally
            variable=self.example_float_var, # the tk variable the widget controls
            length=200
        )
        example_float_var_scale.pack(side=tk.LEFT, padx=10, expand=True)
        ttk.Label(
            template_variable_frame,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
            textvariable=self.example_float_var_display_var, # var truncated to 2 d.p.
            width=5,
            style="Value.TLabel"
        ).pack(side=tk.LEFT)
        tksiril.create_tooltip(example_float_var_scale,
                               "Adjusts the template variable.")

        ###############################
        # Add frame for other variables
        ###############################
        options_frame = ttk.LabelFrame(main_frame, text="Options", padding=10)
        options_frame.pack(fill=tk.X, padx=5, pady=10)

        ##################
        # Example checkbox
        ##################
        self.example_bool_var = tk.BooleanVar(self.root, value=False)
        example_checkbox = ttk.Checkbutton(
            options_frame,
            text="Example checkbox variable",
            variable=self.example_bool_var,
            style="TCheckbutton"
        )
        example_checkbox.pack(anchor=tk.W, pady=2)
        tksiril.create_tooltip(example_checkbox, "Example checkbox.")

        ##########################################################
        # Example file selection
        # using an entry and a callback that triggers a filedialog
        # These file selector widgets have their own frame
        ##########################################################
        file_frame = ttk.Frame(options_frame)
        file_frame.pack(fill=tk.X, pady=5)

        ttk.Label(file_frame, text="File:").pack(side=tk.LEFT)

        self.example_file_path_var = tk.StringVar(self.root, value="")
        example_file_entry = ttk.Entry(
            file_frame,
            textvariable=self.example_file_path_var,
            width=40
        )
        example_file_entry.pack(side=tk.LEFT, padx=(5, 5), expand=True)

        ttk.Button(
            file_frame,
            text="Browse",
            command=self._browse_file,
            style="TButton"
        ).pack(side=tk.LEFT)

        # The template shows examples of sliders and checkboxes
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        # however you can add other sorts of TKinter widgets here

        # Buttons frame
        button_frame = ttk.Frame(main_frame)
        button_frame.pack(pady=20)

        close_btn = ttk.Button(
            button_frame,
            text="Close",
            command=self.close_dialog,
            style="TButton"
        )
        close_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(close_btn,
                               "Close the interface and disconnect from Siril. "
                               "No changes will be made to the current image.")

        apply_btn = ttk.Button(
            button_frame,
            text="Apply",
            command=self.apply_changes,
            style="TButton"
        )
        apply_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(apply_btn,
                               "Apply the script function with the set parameters "
                               "to the current image. Changes can be undone using "
                               "Siril's undo function.")

    def apply_changes(self):
        """ Get the necessary variables from the GUI and call the algorithm """
        try:
            # Get the thread
            with self.siril.image_lock():
                # Get values from the GUI widgets
                example_float_var = self.example_float_var.get()
                example_bool_var = self.example_bool_var.get()
                example_file_path_var = self.example_file_path_var.get()

                # Get current image
                fit = self.siril.get_image()
                fit.ensure_data_type(np.float32)

                # Save original image for undo
                self.siril.undo_save_state(f"Script algorithm: "
                                           "arg={example_float_var:.2f}")

                # Apply script algorithm
                # Your image processing functions go here
                result = template_algorithm(fit, example_float_var, example_bool_var,
                                            example_file_path_var)

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                # Clip and update image data
                fit.data[:] = np.clip(result, 0, 1)
                self.siril.set_image_pixeldata(fit.data)

        except SirilError as e:
            messagebox.showerror("Error", str(e))

    def close_dialog(self):
        """ Close dialog """
        self.root.quit()
        self.root.destroy()

def main():
    """ Main entry point """
    try:
        # Create the GUI interface
        root = ThemedTk()
        TemplateScriptInterface(root)
        root.mainloop()
    except SirilError as e:
        print(f"Error initializing script: {str(e)}", file=sys.stderr)
        sys.exit(1)

if __name__ == "__main__":
    main()

language = "de"
locale_dirs = ["../../translated/"]
gettext_compact = "siril-documentation"
```

Download gui_template.py

The second template is very similar, but provides an alternative interface using a command-line argument vector. This allows the script to be called with arguments using the `pyscript` Siril command: it is useful for scripting algorithms that you might wish to use as part of a bigger scripted workflow.

Quellcode 2: gui_and_args_template.py

```
# Template for Siril script with a GUI
# SPDX-License-Identifier: # insert license identifier here
# insert author's name and details here

"""
This script template provides an exemplar for use in constructing sirilpy
scripts that provide a Tkinter GUI but which can also be called using an
argument vector (i.e. so that they can be scripted for use with the Siril
pyscript command). A similar template exists for pure GUI scripts, which
is slightly simpler owing to not having to accommodate command-line
arguments.
"""

# Core module imports
import os
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
import sys
import math
import argparse
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import numpy as np

import sirilpy as s
from sirilpy import tksiril, SirilError
s.ensure_installed("ttkthemes")
# Add other calls to ensure_installed() here to ensure that any non-core
# modules that are required by the script are installed and available
# to be imported.

# Note: pylint will complain that the following imports are not at the
# top of the module. This deviation from python style is required in order
# to run ensure_installed() to ensure that any non-core modules are
# available before attempting to import them.
from ttkthemes import ThemedTk
# Add any additional imports here

VERSION = "1.0.0"

def template_algorithm(fit, example_float_var,
                       example_bool_var,
                       example_file_path_var):
    """
   Template processing function, insert your algorithm here.
   This is the non-GUI part of the script and is entirely up to you to write.
   Although presented in the template as a single method it may comprise
   classes, methods and anything else.
   """
    print(f"Argument: {example_float_var}")
    print(f"Bool var: {example_bool_var}")
    print(f"Bool var: {example_file_path_var}")
    return fit

class TemplateScriptInterface:
    """ This class provides the GUI and related callbacks """
    def __init__(self, root=None, cli_args=None):
        # If no CLI args, create a default namespace with defaults
        if cli_args is None:
            parser = argparse.ArgumentParser()
            parser.add_argument("-example_float_var", type=float, default=1.0)
            parser.add_argument("-example_bool_var", type=bool, default=False)
            parser.add_argument("-example_file_path_var", type=str, default="")
            # add other arguments to match the ones in main()
            cli_args = parser.parse_args([])

        self.cli_args = cli_args
        self.root = root

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        if root:
            self.root.title(f"Template Script Interface - v{VERSION}")
            self.root.resizable(False, False)
            self.style = tksiril.standard_style()

        # Initialize Siril connection
        self.siril = s.SirilInterface()

        try:
            self.siril.connect()
        except s.SirilConnectionError:
            if root:
                self.siril.error_messagebox("Failed to connect to Siril")
            else:
                print("Failed to connect to Siril")
            return

        # Initial checks: example - check if an image is loaded
        if not self.siril.is_image_loaded():
            if root:
                self.siril.error_messagebox("No image is loaded")
            else:
                print("No image is loaded")
            return

        # Check if the version of Siril is high enough
        # 1.3.6 is the baseline when python scripting became available but in time
        # you may need to check for a higher version number in order to use newer
        # features.
        try:
            self.siril.cmd("requires", "1.3.6")
        except s.CommandError:
            return

        # Create the UI and match its theme to Siril
        if root:
            self.create_widgets()
            tksiril.match_theme_to_siril(self.root, self.siril)

        # Only apply changes if CLI arguments are non-default
        if cli_args and (cli_args.example_float_var != 1.0):
            self.apply_changes(from_cli=True)

    # This function is used to round values to n decimal places, it is
    # always useful if you have sliders in your UI.
    def _floor_value(self, value, decimals=2):
        """Floor a value to the specified number of decimal places"""
        factor = 10 ** decimals
        return math.floor(value * factor) / factor

    # Here is an example function to round the displayed value of a
    # variable to 2 decimal places. You will need one of these for each
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    # slider control you have.

    def _update_example_float_variable_display(self, *args): # pylint: disable=unused-argument
        """Update the displayed template variable value with floor rounding"""
        value = self.example_float_var.get()
        rounded_value = self._floor_value(value)
        self.example_float_var_display_var.set(f"{rounded_value:.2f}")

    def _browse_file(self):
        """
        Use a TK filedialog to browse for a file. Note that this callback is
        specific to the variable being updated; you might need to have several
        similar callbacks if you have more than one file selection widget.
        """
        filename = filedialog.askopenfilename(
            title="Select File",
            initialdir=os.path.expanduser("~")
        )
        if filename:
            self.example_file_path_var.set(filename)

    def create_widgets(self):
        """Create the GUI's widgets, connect signals etc. """
        # Main frame with no padding
        main_frame = ttk.Frame(self.root)
        main_frame.pack(fill=tk.BOTH, expand=True, padx=0, pady=0)

        # Title
        title_label = ttk.Label(
            main_frame,
            text="Template Script",
            style="Header.TLabel"
        )
        title_label.pack(pady=(0, 20))

        # Parameters frame
        params_frame = ttk.LabelFrame(main_frame, text="Parameters", padding=10)
        params_frame.pack(fill=tk.X, padx=5, pady=5)

        ##########################################################
        # Example control with an embedded variable in its own frame
        ##########################################################

        template_variable_frame = ttk.Frame(params_frame)
        template_variable_frame.pack(fill=tk.X, pady=5)

        ttk.Label(template_variable_frame,
            text="Example floting point parameter:").pack(side=tk.LEFT)
        # This is the actual float variable
        self.example_float_var = tk.DoubleVar(self.root,
            value=self.cli_args.example_float_var)
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        # This is a string representation of the float var limited to 2 decimal places
        # It is used in the scale label
        self.example_float_var_display_var = tk.StringVar(
            self.root,
            value=f"{self._floor_value(self.cli_args.example_float_var):.2f}")
        # Add trace to update display when slider changes
        self.example_float_var.trace_add("write",
            self._update_example_float_variable_display)

        example_float_var_scale = ttk.Scale(
            template_variable_frame,
            from_=0.0, # set your range minimum here
            to=1.0,    # set your range maximum here
            orient=tk.HORIZONTAL, # oriented horizontally
            variable=self.example_float_var, # the tk variable the widget controls
            length=200
        )
        example_float_var_scale.pack(side=tk.LEFT, padx=10, expand=True)
        ttk.Label(
            template_variable_frame,
            textvariable=self.example_float_var_display_var, # var truncated to 2 d.p.
            width=5,
            style="Value.TLabel"
        ).pack(side=tk.LEFT)
        tksiril.create_tooltip(example_float_var_scale,
            "Adjusts the template variable.")

        ################################
        # Add frame for other variables
        ################################
        options_frame = ttk.LabelFrame(main_frame, text="Options", padding=10)
        options_frame.pack(fill=tk.X, padx=5, pady=10)

        ##################
        # Example checkbox
        ##################
        self.example_bool_var = tk.BooleanVar(self.root,
            value=self.cli_args.example_bool_var
        )
        example_checkbox = ttk.Checkbutton(
            options_frame,
            text="Example checkbox variable",
            variable=self.example_bool_var,
            style="TCheckbutton"
        )
        example_checkbox.pack(anchor=tk.W, pady=2)
        tksiril.create_tooltip(example_checkbox, "Example checkbox.")

        ##########################################################
        # Example file selection
        # using an entry and a callback that triggers a filedialog
        ##########################################################
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        file_frame = ttk.Frame(options_frame)
        file_frame.pack(fill=tk.X, pady=5)

        self.example_file_path_var = tk.StringVar(self.root,
            value=self.cli_args.example_file_path_var)
        example_file_entry = ttk.Entry(
            file_frame,
            textvariable=self.example_file_path_var,
            width=40
        )
        example_file_entry.pack(side=tk.LEFT, padx=(0, 5), expand=True)

        ttk.Button(
            file_frame,
            text="Browse",
            command=self._browse_file,
            style="TButton"
        ).pack(side=tk.LEFT)

        # The template shows examples of sliders and checkboxes
        # however you can add other sorts of TKinter widgets here

        # Buttons frame
        button_frame = ttk.Frame(main_frame)
        button_frame.pack(pady=20)

        close_btn = ttk.Button(
            button_frame,
            text="Close",
            command=self.close_dialog,
            style="TButton"
        )
        close_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(close_btn,
                               "Close the interface and disconnect from Siril. No "
                               "changes will be made to the current image.")

        apply_btn = ttk.Button(
            button_frame,
            text="Apply",
            command=self.apply_changes,
            style="TButton"
        )
        apply_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(apply_btn,
                               "Apply the script function with the set parameters "
                               "to the current image. Changes can be undone using "
                               "Siril's undo function.")

    def apply_changes(self, from_cli=False):
        """
        Get the necessary variables from CLI args or the GUI and call the algorithm
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
        """
        try:
            # Get the thread
            with self.siril.image_lock():
                # Determine parameters: prefer CLI args if provided,
                # else use GUI values
                if from_cli and self.cli_args:
                    example_float_var = self.cli_args.example_float_var
                    example_bool_var = self.cli_args.example_bool_var
                    example_file_path_var = self.cli_args.example_file_path_var
                else:
                    example_float_var = self.example_float_var.get()
                    example_bool_var = self.example_bool_var.get()
                    example_file_path_var = self.example_file_path_var.get()

                # Get current image
                fit = self.siril.get_image()
                fit.ensure_data_type(np.float32)

                # Save original image for undo
                self.siril.undo_save_state(f"Script algorithm: "
                                           "arg={example_float_var:.2f}")

                # Apply script algorithm
                # Your image processing functions go here
                result = template_algorithm(fit, example_float_var,
                                            example_bool_var,
                                            example_file_path_var)

                # Clip and update image data
                fit.data[:] = np.clip(result, 0, 1)
                self.siril.set_image_pixeldata(fit.data)

            if from_cli:
                print("Script algorithm applied successfully.")

        except SirilError as e:
            if from_cli:
                print(f"Error: {str(e)}")
            else:
                messagebox.showerror("Error", str(e))

    def close_dialog(self):
        """ Close dialog """
        if hasattr(self, 'root'):
            self.root.quit()
            self.root.destroy()

def main():
    """ Main entry point """
    parser = argparse.ArgumentParser(description="Template python script")
    parser.add_argument("-example_float_var", type=float, default=1.0,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
                        help="Describe the variable here (0.0 to 1.0)")
    parser.add_argument("-example_bool_var", type=bool, default=False,
                        help="Describe the variable here (bool, default is False)")
    parser.add_argument("-example_file_path_var", type=str, default="",
                        help="Describe the variable here (example file path)")

    args = parser.parse_args()

    try:
        if any([args.example_float_var != 1.0, args.example_bool_var is not False]):
            # CLI mode
            TemplateScriptInterface(cli_args=args)
        else:
            # GUI mode
            root = ThemedTk()
            TemplateScriptInterface(root)
            root.mainloop()
    except SirilError as e:
        print(f"Error initializing script: {str(e)}", file=sys.stderr)
        sys.exit(1)

if __name__ == "__main__":
    main()

language = "de"
locale_dirs = ["../../translated/"]
gettext_compact = "siril-documentation"
```

Download gui_and_args_template.py

20.7 Current list of Python scripts for Siril 1.4

This page is the first shot at getting a list of Python scripts that is somewhat intelligible and searchable. It was done manually and has only a very succinct description for each script, but will be improved with more words for each with time.

20.7.1 Image processing section

Scripts that process a single image for visual enhancements, in general a stack result.

SyQon/Prism.py	SyQon Prism AI denoise. This is a paying product
SyQon/Starless.py	SyQon Starless AI star removal. This is a paying product
SyQon/Parallax.py	SyQon Parallax AI: aberration correction, star reduction and sharpening. This is a paying product
SyQon/SyQon_Studio_Complete.	New SyQon script that will group all paid models (Parallax, Prism, Starless)
RC-Astro/NoiseXTerminator.py	RC Astro AI denoise. This is a paying product
RC-Astro/StarXTerminator.py	RC Astro Starless AI star removal. This is a paying product
RC-Astro/BlurXTerminator.py	RC Astro Deconv AI: aberration correction, star reduction and sharpening. This is a paying product
processing/Statistical_Stretch.py	Seti Astro Statistical Stretch

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

processing/AberrationRemover.py	Aberration Remover AI
processing/AstroColorMixer.py	multi-pass, hue-targeted color and luminance refinement for nonlinear RGB data.
processing/ContinuumSubtraction.py	Narrowband Continuum Subtraction script
processing/DeepSNR.py	DeepSNR AI denoise (wrapper around the third-party DeepSNR tool)
processing/StarNet.py	StarNet 2.5 star removal (wrapper around the third-party StarNet tool), with optional star mask generation
processing/GPS_Process.py	Image processing based on CosmicClarity and GraXpert (background extraction, denoise and/or sharpening)
processing/Narrowband_Palette_Picker.p	Helps preview different palettes for narrowband image composition
processing/GraXpert-AI.py	Integration of the GraXpert gradient removal tool (AI)
processing/CosmicClarity_Denoise.py	Cosmic Clarity Denoise
processing/DBXtract.py	Extract the Sulfur II, Hydrogen Alpha and Oxygen III signal from dual band filters to compose SHO images in color cameras
processing/Hubble_Palette_from_Band_OSC.py	Create different "Hubble-like" palettes from OSC (One-Shot Color) images acquired with a dual-band Ha/OIII filter
processing/PalettePicker.py	Script that help user to create its own palette for narrowband filters
processing/NarrowbandNormalization.p	Script that allows easy narrowband normalization on monochrome SHO combined images and on OSC dual band images
processing/HDR_multiscale.py	Wavelet-Based Dynamic Range Compression
processing/HDR_Blender.py	Inverse Variance Maximum Likelihood Estimate (MLE) HDR Blender. Blends multiple images with different exposures
processing/NB_2_RGB.py	Monochrome image combination into a color image
processing/SCUNet_Denoise.py	SCUNet image denoiser (AI)
processing/CosmicClarity_Superres.py	Cosmic Clarity Superres
processing/CosmicClarity_Native.py	Cosmic Clarity AI-powered sharpening, denoising, super resolution and star removal
processing/AutoBGE.py	Auto Background Extraction script
processing/AutoGradientRemoval.py	Auto Gradient Removal tool. This script will be implemented in the Siril code in the next version
processing/CosmicClarity_Satellite.py	Cosmic Clarity Satellite Removal
processing/CosmicClarity_Darkstar.py	Cosmic Clarity Darkstar
processing/ERBill_Star_Reduction.py	Script for reducing stars using pixel math
processing/DSAStar_Reduction.py	Script for reducing stars using pixel math
processing/CosmicClarity_Sharpen.py	Cosmic Clarity sharpening process
VeraLux/VeraLux_Alchemy.py	Linear-Phase Narrowband Normalization & Mixing
VeraLux/VeraLux_Nox.py	Physically-Faithful Photometric Gradient Reduction
VeraLux/VeraLux_StarComposer.py	High-Fidelity Star Reconstruction Engine
VeraLux/VeraLux_Revela.py	Photometric Local Contrast & Texture Engine
VeraLux/VeraLux_HyperMetric_Stret	Photometric Hyperbolic Stretch Engine

Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

VeraLux/VeraLux_Starting_Point.py	Interactive Workflow Guide & Manual
VeraLux/VeraLux_Curves.py	Spline-Based Photometric Sculpting Engine
VeraLux/VeraLux_Silentium.py	Linear-Phase Noise Suppression Engine
VeraLux/VeraLux_Vectra.py	Vector Color Grading & Chromatic Surgery Engine
processing/InteractiveHaloEditor.py	Edit halos, gradients, and artifacts using an unlimited stack of smooth ellipse masks, freehand brushes, and custom spline curves
processing/Lighthouse.py	Applies Lightroom-style adjustments

20.7.2 Sequence pre-processing section

Scripts that process a set of images, an alternative to usual preprocessing scripts.

preprocessing/AMSP.py	Siril Wizard – Automatic Multi-Session Processing. Drag and drop all files and it does the stacks
preprocessing/StorageFriendlyStacking.py	Storage Friendly Stacking Script
preprocessing/Naztronomy-Smart_Telescope_PP.py	Naztronomy - Smart Telescope Preprocessing script
preprocessing/GPS_Preprocess.py	Preprocessing with multi-session, mosaics and many optional steps support
preprocessing/ER-CometStartrail.py	Stack a startrail from a sequence
preprocessing/osc-multi-night-stacking-v1.2.py	Multi-night stacking
preprocessing/Naztronomy-OSC_PP.py	Naztronomy - OSC Image Preprocessing script with mosaics support
preprocessing/Naztronomy-Mono_PP.py	Naztronomy - Monochrome Image Preprocessing script with mosaics support

20.7.3 Utility section

Scripts that provide some tool for data analysis or other non-processing related tasks.

core/Siril_Catalog_Installer.py	Siril Catalog Installer
core/GPU_Manager.py	A GUI tool for managing ONNX, PyTorch, and JAX installations
utility/Selected_Star_Spectrum.py	Plots the Gaia DR3 continuous spectrum for any selected Gaia DR3 source
utility/RegistrationInspector.py	Displays the frames of selected images to check framing
utility/SuperStack.py	Performs superstacking (moving average)
utility/plot3D.py	Plots the current image or selection in 3D using matplotlib
utility/Sequence_Statistics_Analyzer.p	Analyze frames and plot key statistics
utility/Workflow_Companion.py	Deep Space Astro drag-and-drop workflow companion: queue, reorder and launch scripts and Siril functions
utility/Workflow_Summarizer.py	Uses Google Gemini to summarize the Siril log into readable workflow documentation
utility/Blink_Browse_Filter_Sort.py	Image browser / filter / sorter with adaptive caching

Fortsetzung auf der nächsten Seite

Tab. 2 – Fortsetzung der vorherigen Seite

utility/Svenesis-GradientAnalyzer.py	Gradient Analyzer
utility/Svenesis-MultipleHistogramViewer.py	Multiple Histogram Viewer
utility/Svenesis-CosmicDepth3D.py	CosmicDepth 3D
utility/Patch_Inpainting_Tool.py	Corrects defects and creates a star mask
utility/Svenesis-AnnotateImage.py	Full size image annotation
utility/AF_Multi_Crop.py	Multi-crop script
utility/Svenesis-BlinkComparator.py	Blink Comparator
utility/Satellite_Trail_Removal.py	Remove satellite trails
utility/Sequence_Deleter.py	GUI to delete sequences
utility/ImageWindow.py	Pseudo-MDI GUI for storing and swapping images
utility/Dwarfium_Archive_Selector.py	Prepare sessions from Dwarf telescopes
utility/AstroT3kFetch.py	Fetch and classify astrophotography frames
utility/Diffraction_Spike_Overlay.py	Add diffraction spikes
utility/Flat_On_Flat_Analyzer.py	Analyze flat field effectiveness
utility/Distortion3D.py	Plot 3D distortion map
utility/Autocrop.py	Autocrop stacked images
utility/Signature_Tool.py	Insert a signature/logo
utility/Galaxy_Annotations.py	Create galaxy annotations from Simbad queries
utility/Asteroid_Comet_Finder.py	Search for asteroids and comets
utility/AutoStretch_Preview.py	Interactive AutoStretch preview
utility/HertzsprungRussell.py	Create a Hertzsprung-Russell diagram

Warnung

Not all scripts are written by, or the responsibility of, the Siril development team!

With the introduction of Python scripting the power available to Siril script writers has increased enormously. With great power comes great responsibility!

You, as the user, need to ensure you trust the authors of scripts you use. 1.4.0 introduces the script repository. Some of the scripts there have been written by the Siril development team but others have been written by third party contributors. Please read the Key Info section regarding guidance for script authors, end users and the rules for the script repository.

20.8 Skripte laden

A small number of basic script files come packaged with Siril, however you can add more from the script repository. To do so, go to the *Scripts menu -> Get Scripts*. This will open the *Scripts* tab of the *Preferences* dialog. Here, below the list of local script directories, you will find a list of scripts in the repository.

The scripts can be sorted by category, script name, type or by whether or not they are selected.

As there may eventually be a large number of scripts, not all of which will be relevant to your workflow, you can configure the scripts you want to appear in the *Scripts* menu. Simply select the checkbox next to them and click Apply to update the menu.

Tipp

You don't need to select scripts to appear in the menu for them to be usable with the `pyscript` command: this will search all the local script directories as well as the whole of the local scripts repository.

Warnung

Script authors may submit updates. Occasionally, inevitably, there may be a bad update that breaks the script for you. Bugs need to be reported to the script author: for scripts written by the Siril team you can report them at the scripts GitLab repository and for those written by third party contributors you should report them at the contact details in the script.

The script can be opened in the script editor by double clicking on it in the list. This is important for several reasons:

- It allows you to find the details of the script author for bug reporting.
- When you double click the script there is an option to revert to previous versions.
- This means that if an update broke the script, you can open the previous version and save a local copy in one of your local script folders for use until the script author fixes the bad update. (The version saved in a local folder will take priority over a repository version with the same name, but it will appear in a different subfolder in the menu with the name of your local scripts folder.)

KAPITEL 21

Headless Modus

Siril kann sowohl mit seiner grafischen Benutzeroberfläche (GUI) als auch mit einer Befehlszeilenschnittstelle (CLI) arbeiten, für die nicht einmal ein Bildschirm erforderlich ist. Es kann Bilder für andere Programme auf Remote-Computern oder oder in virtuellen Maschinen verarbeiten, indem es entweder Skripte oder in Echtzeit generierte Operationen, sogenannte Befehle, verwendet. Die Fähigkeiten des "headless Siril" sind in der Tat die der *verfügbaren Befehle*. Es gibt mehr als hundert Befehle, mit denen die Kalibrierung, die Verarbeitung und die photometrische Analyse automatisch durchgeführt werden können.

Befehle können auch in der GUI-Version von Siril verwendet werden, entweder über die eingebettete Befehlszeile am unteren Rand des Bedienfelds oder mit *Skripte*. Skripte sind einfach eine Textdatei, die eine Liste von Befehlen enthält. Es wird empfohlen, die Seite *Skripte* zu lesen, bevor Sie fortfahren.

In der Headless-Version können Befehle entweder durch Übergabe eines Skripts ausgeführt werden, oder indem die Standardeingabe als Skript gesetzt wird und Befehle in diese geschrieben werden, mit der `-s -` Kommandozeilenoption, oder indem benannte Pipes (named Pipes) verwendet werden.

Hier ist ein Beispiel für einen bash-Code, der den Headless-Modus aufruft, der den Master-Bias aufbaut und das Hintergrundrauschen in roter Farbe auf der Konsole ausgibt:

```
#!/bin/bash
# bash commands to prepare files

initdir=$(pwd)

######## Set your own variables #############
SCRIPTS_DIRECTORY=$initdir
SIRIL_PATH=siril-cli
#############################################

# Removing process folder if exists #
rm -rf $SCRIPTS_DIRECTORY/process

echo "Running siril bash script in $initdir"
output=$($SIRIL_PATH -d $SCRIPTS_DIRECTORY -s - <<ENDSIRIL
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
##########################################
#
# Example of script that create a master-bias
# and print in red the noise of the image
#
##########################################

requires 1.0.0

# Convert Bias Frames to .fit files
cd biases
convert bias -out=../process
cd ../process

# Stack Bias Frames to bias_stacked.fit
stack bias rej 3 3 -nonorm -out=master-bias
cd ../..

close
ENDSIRIL
)

log_line=$(echo "$output" | grep -o "log: Background noise value.*")
echo -e "\e[31m$log_line\e[0m"

echo done Siril part
```

21.1 Befehlsstrom (Pipe)

Beginnend mit Version 0.9.10 wurde ein neuer Modus eingeführt, in dem Befehle über eine "Named Pipe" gesendet und Logdateien und Status über eine andere abgefragt werden können. Dieser Modus wird mit dem Befehlszeilenargument -p aktiviert.

Das Protokoll ist recht einfach: Siril empfängt Befehle und gibt einige Aktualisierungen aus. Nur Befehle, die als skriptfähig gekennzeichnet sind, können mit diesem System verwendet werden. Immer wenn die Befehlseingangs-Pipe geschlossen oder der Befehl *cancel* ausgeführt wird, wird der laufende Prozess beendet, so als ob die Schaltfläche *Stop* in der Benutzeroberfläche angeklickt worden wäre. Die Pipes heißen `siril_command.in` und `siril_command.out` und sin im Verzeichnis `/tmp` auf Unix-basierten systemen verfügbar. Seit Version 1.1.0, können die Pfadnamen der Pipes mit den Optionen `-r``und ``-w` konfiguriert werden, was externen Programmen erlaubt, sie vor dem Start von Siril zu erstellen, typischerweise mit dem Befehl `mkfifo`. Ebenfalls neu in 1.1.0 ist, dass der Ping-Befehl `ping` einfach einen Status zurückgibt, der angibt, ob Siril bereit ist, einen Befehl zu verarbeiten oder bereits beschäftigt ist.

Die Ausgaben von Siril ist eine Folge einzeiliger Texte und wie folgt formatiert:

- `ready` wird beim Start ausgegeben um anzuzeigen, dass Siril bereit ist Befehle zu verarbeiten
- `log:` gefolgt von einer Logmeldung
- `status: verb [subject]`, wobei verb entweder `starting`, `success`, `error` oder `exit` sein kann (exit-Nachrichten sind noch nicht implementiert). Das Subjekt ist der Name des aktuellen Befehls, mit Ausnahme von exit, das anzeigt, dass siril einen fatalen Fehler erlitten hat und beendet werden muss.
- `progress: value%` ist die Entsprechung eines Fortschrittsbalkens, es werden periodisch Prozente und manchmal 0% am Ende einer Verarbeitung als Leerlaufinformation gesendet.

KAPITEL 22

Pfadnamen Parsing

Parsing ist die Fähigkeit zu parsen, d.h. Zeichenketten auf der Grundlage der im *FITS*-Header enthaltenen Daten zu schreiben. Die Pfadanalyse, die in Siril 1.2.0 eingeführt wurde, zielt darauf ab, der Skripterstellung mehr Flexibilität zu verleihen, indem Header-Daten zum Schreiben/Lesen von Dateinamen oder Pfaden verwendet werden können. Zur Zeit wird dies mit den folgenden Befehlen verwendet:

- ref: "Kalibrieren <calibrate>
- *stack* and *stackall*
- Alle *saveXXX*-Befehle

und natürlich deren GUI-Entsprechungen.

22.1 Beispiel für die Syntax

Beginnen wir mit einem einfachen Beispiel. Angenommen, Sie haben eine Datei mit dem Namen "light_00001.fit" und möchten in Ihrer Masterdark-Bibliothek ein Masterdark finden, das den Eigenschaften dieses Lights entspricht. Da Sie eine Konvention zur Benennung Ihres Masterdarks gewählt haben, wissen sie, dass der korrekte Dateiname des Darks etwas so heißen sollte:

```
DARK_"exposure"s_G"gain"_O"offset"_T"temperature"C_bin"binning".fit
```

wobei die Begriffe in den Anführungszeichen durch die Werte aus dem FITS-Header ihres Lights ersetzt werden. Bei einer Belichtungszeit von 120s, einer Sensortemperatur von -10°C, einem Gain/Offset von 120/30 und Binning 1 würde das Masterdark so heißen:

```
DARK_120s_G120_O30_T-10C_bin1.fit
```

Nun, das ist genau das, was diese Funktion ermöglicht. Wenn Sie den Namen des Darks mit den gerade erläuterten Konventionen angeben, können Sie Siril anweisen, das Light zu öffnen, seinen Header zu lesen und die Header-Werte zu verwenden, um eine solche Zeichenkette zu erstellen (und sie dann zu verwenden, um das Light mit dem passenden Masterdark vorzuverarbeiten).

Sie können die im Header enthaltenen Informationen entweder über den Befehl *dumpheader* oder über das Werkzeug lesen: *Werkzeuge → Extras → FITs Header...* ` Normalerweise erhalten Sie einen Ausdruck wie den folgenden (der Übersichtlichkeit halber wurden einige Schlüssel entfernt):

```
SIMPLE  =                    T / C# FITS
BITPIX  =                   16 /
NAXIS   =                    2 / Dimensionality
NAXIS1  =                 4144 /
NAXIS2  =                 2822 /
BZERO   =                32768 /
EXTEND  =                    T / Extensions are permitted
IMAGETYP= 'LIGHT'              / Type of exposure
EXPOSURE=                120.0 / [s] Exposure duration
DATE-OBS= '2022-01-24T01:03:34.729' / Time of observation (UTC)
XBINNING=                    1 / X axis binning factor
YBINNING=                    1 / Y axis binning factor
GAIN    =                  120 / Sensor gain
OFFSET  =                   30 / Sensor gain offset
INSTRUME= 'ZWO ASI294MC Pro'   / Imaging instrument name
SET-TEMP=                -10.0 / [degC] CCD temperature setpoint
CCD-TEMP=                -10.0 / [degC] CCD temperature
BAYERPAT= 'RGGB'               / Sensor Bayer pattern
TELESCOP= '61EDPH'             / Name of telescope
FILTER  = 'DualBand'           / Active filter name
OBJECT  = 'Rosette Nebula     '/ Name of the object of interest
OBJCTRA = '06 30 36'           / [H M S] RA of imaged object
OBJCTDEC= '+04 58 51'          / [D M S] Declination of imaged object
RA      =     97.6960081674312 / [deg] RA of telescope
DEC     =     4.99212765957446 / [deg] Declination of telescope
END
```

Das Format für die Angabe des spezifischen Dark-Dateinamens würde dann lauten:

```
DARK_$EXPTIME:%d$s_G$GAIN:%d$_O$OFFSET:%d$_T$SET-TEMP:%d$C_bin$XBINNING:%d$.fit
```

Alle zu parsenden Teile weden wie folt gebildet: $**SCHLÜSSEL**:**fmt**$

- **SCHLÜSSEL** ist jeder beliebige (gültige) Schlüssel aus dem FITS-Header des Lights
- **fmt** ist ein format specifier.

Zum Beispiel wird $EXPTIME:%d$ zu *120* geparsed, wenn das Light 120s lang belichtet wurde. Wenn sie $EXPTIME:%0.1f$ angeben, ergibt sich durch die Formatangabe *%x.yf* jedoch *120.0*.

Der gesamte obige Ausdruck würde also zu folgendem Ergebnis führen:

```
DARK_**120**s_G**120**_O**30**_T**-10**C_bin**1**.fit
```

In diesem ersten Beispiel haben wir nur die Umwandlung in Ganzzahlen mit *%d* verwendet. Es gibt aber auch andere gebräuchliche Formatierer, die Sie verwenden können:

- `%x.yf` für Fließkommazahlen
- `%s` für Zeichenketten

Bemerkung

Bei Zeichenketten werden führende und nachfolgende Leerzeichen immer entfernt, während Leerzeichen innerhalb einer Zeichenkette durch _-Zeichen ersetzt werden. Beispiel: `$OBJECT:%s$` würde umgewandelt zu `Rosette_Nebula`.

Sie können auch einige weniger gebräuchliche Formatierer verwenden:

- Um ein Datum aus einem Datums-/Uhrzeit-Headerfeld zu parsen, können Sie den speziellen Nicht-Standard-Formatierer **dm12** verwenden, was "Datum - 12 Stunden" bedeutet. In der obigen Headerzeile hat der Schlüssel DATE-OBS den Wert `2022-01-24T01:03:34.729`. `$DATE-OBS:dm12$` würde diesen Wert in `2022-01-23` umwandeln, was dem Beginn der Nacht entspricht. Sie können auch den speziellen Formatierer **dm0** verwenden, der nur das Datum parst, ohne 12 Stunden zu subtrahieren.
- Um ein Datum aus einem Datums-/Uhrzeit-Headerfeld zu parsen, können Sie den speziellen Nicht-Standard-Formatierer **dt** verwenden, was einfach Datum+Zeit bedeutet. Im obigen Header enthält der Schlüssel `DATE-OBS` den Wert `2022-01-24T01:03:34.729`. `$DATE-OBS:dt$`. $DATE-OBS:dt$ wird das zu `2022-01-24_01-03-34` konvertieren.
- Um „RA“- und „DEK“-Informationen aus den Header-Schlüsseln „OBJCTRA“ und „OBJCTDEC“ zu analysieren, können Sie die speziellen, nicht standardmäßigen Formatierer **ra** und **dec* verwenden. *. Im obigen Header haben die Schlüssel „OBJCTRA“ und „OBJCTDEC“ den Wert „06 30 36“ bzw. „+04 58 51“. „$OBJCTRA:ra$_$OBJCTDEC:dec$“ würde in „06h30m36s_+04d58m51s“ umgewandelt werden.
- Um „RA“- und „DEK“-Informationen aus den Header-Schlüsseln „RA“ und „DEC“ im Dezimalformat zu analysieren, können Sie die speziellen, nicht standardmäßigen Formatierer **ran** verwenden. und **decn**. Im obigen Header haben die Schlüssel „RA“ und „DEC“ den Wert „97.6960081674312“ bzw. „4.99212765957446“. „$RA:ran$_$DEC:decn$“ würde in „06h30m47s_+04d59m32s“ umgewandelt werden.

Ein gutes Beispiel für den Ergebnisdateinamen des Stackings ist der folgende Ausdruck:

```
$OBJECT:%s$_$FILTER:%s$_$STACKCNT:%d$x$EXPTIME:%d$sec_G$GAIN:%d$_O$OFFSET:%d$_T$CCD-TEMP:
↪%d$°C_$DATE-OBS:dm12$
```

die etwa wie:

```
NGC_7023_L_57x120sec_G100_O50_T-9°C_2023-10-07
```

Um die Syntax zu testen, können Sie ein Bild laden und den Befehl *parse* verwenden, wie unten gezeigt.

Siril Kommandozeile

```
parse str [-r]
```

Parsed the String **str** basierend auf den Informationen im Header des aktuelle geladenen Bildes. Hauptzweck dieses Befehls ist es, das Path-Parsing von Header-Schlüsseln zu debuggen welches in anderen Befehlen genutzt werden kann.
Option **-r** specifies the string is to be interpreted in read mode. In read mode, all wildcards defined in string **str** are used to find a file name matching the pattern. Otherwise, default mode is write mode and wildcards, if any, are removed from the string to be parsed.

Wenn **str** mit dem Präfix *$def* beginnt, wird es als reserviertes Schlüsselwort erkannt und in den in gui_prepro.dark_lib, gui_prepro.flat_lib, gui_prepro.bias_lib oder gui_prepro.stack_default gespeicherten Strings

für *$defdark*, *$defflat*, *$defbias* bzw. *$defstack* gesucht.
Das Schlüsselwort *$seqname$* kann auch verwendet werden, wenn eine Sequenz geladen ist

22.2 Auffinden einer Datei mit Pfadanalyse

Im obigen Beispiel haben wir gesehen, dass wir den Namen eines Masterdarks anhand der Informationen aus dem FITS-Header des zu kalibrierenden Lights finden können. Dies wird im Befehl `parse` der `Lesemodus` genannt.

Dieses Verhalten wird hauptsächlich in Verbindung mit dem Befehl/Tab *calibrate* und zum Suchen von Master-Verzerrungsdateien verwendet. In der Option `-dark=` des Befehls oder im Feld `Dark` der GUI können Sie die oben beschriebene Syntax verwenden. Damit können Sie sicher sein, dass das passende Dark zum Kalibrieren des Lights abgerufen wird. Dasselbe gilt gleichermaßen für `Bias` und `Flat`. Sie können natürlich auch einen vollständigen (oder relativen) Pfad zur Datei angeben. Und der Pfad kann auch Ausdrücke derselben Art enthalten.

Bei Flats zum Beispiel möchten sie vielleicht den Pfad zu einer Bibliothek angeben, die Filter- oder Teleskopinformationen enthalten könnte, da Sie möglicherweise mehrere Setups haben. Ein Pfad wie:

```
~/astro/masters/flats/$INSTRUME:%s$_$TELESCOP:%s$/$FILTER:%s$/FLAT_bin$XBINNING:%d$.fit
```

Würde dann umgewandelt in:

```
~/astro/masters/flats/ZWO_ASI294MC_Pro_61EDPH/DualBand/FLAT_bin1.fit
```

und ist ein gültiger Wert für `Flat`.

Wenn Sie dies jedes Mal, wenn Sie kalibrieren, als Befehl in Ihre Skripte oder in das Feld `Flat` der grafischen Benutzeroberfläche schreiben müssten, könnte das natürlich etwas mühsam werden. In diesem Fall kommen reservierte Schlüsselwörter zu Hilfe. Es gibt 3 reservierte Schlüsselwörter für Master:

- `$defdark`
- `$defflat`
- `$defbias`
- `$defdisto`

Ihre Werte werden im Abschnitt *Einstellungen* → *Vorverarbeitung* <preferences/preferences_gui:pre-processing> festgelegt. Sie können sie auch über Skripting mit dem Befehl *set* festlegen. Sie entsprechen den Werten von `gui_prepro.dark_lib`, `gui_prepro.flat_lib`, `gui_prepro.bias_lib` und `gui_prepro.disto_lib`.

Wenn diese Werte festgelegt wurden und Sie sie als Standardwerte verwenden möchten, werden Sie in den Feldern der Registerkarte *Kalibrierung* angezeigt. Sie können auch beginnen, Ihre Skripte unter Verwendung dieser Schlüsselwörter zu schreiben. Der Kalibrierungsschritt eines neuen generischen Skripts für eine Farbkamera könnte z.B. folgendermaßen aussehen:

```
calibrate light -dark=$defdark -cc=dark -flat=$defflat -cfa -equalizecfa -debayer
```

Auf diese Weise werden die Master-Dateien direkt aus den Bibliotheken ausgewählt, und es wird sichergestellt, dass sie bei der Kalibrierung nicht vertauscht werden.

22.3 Schreiben einer Datei mit Dateinamen-Parsing

Es ist zwar praktisch, auf diese Weise Dateien finden zu können, aber es wäre ebenso nützlich, diese Syntax zu verwenden, um Dateien während des Stackings zu speichern. Genau darum geht es beim `Schreibmodus`. Die Syntax kann im Feld `-out=` der Befehle *stack* und *stackall*, oder im entsprechenden Eingabefeld der GUI benutzt werden.

Nehmen wir an, Sie wollen ein allgemeines Skript schreiben, welches bei jedem Update Ihrer Bibliothek die Master-Darks vorbereitet. In der Zeile `stack` des Skripts können Sie dann schreiben:

```
stack dark rej 3 3 -nonorm -out=$defdark
```

Diese Zeile stellt sicher, dass das resultierende Masterdark an der richtigen Stelle mit dem richtigen Dateinamen gespeichert wird, der dann zur Kalibrierung der Lights abgerufen werden kann.

Um noch mehr Flexibilität mit dem `stack`-Befehl zu erreichen, gibt es zwei weitere reservierte Schlüsselwörter:

- `$defstack`
- `$sequencename$`

Wie bei den Standard-Mastern wird `$defstack` im gleichen Abschnitt der Voreinstellungen konfiguriert, oder mit dem Befehl *set* auf `gui_prepro.stack_default`. Nehmen wir zum Beispiel an, Sie haben `$defstack` definiert als:

```
Result_$OBJECT:%s$_$DATE-OBS:dm12$_$LIVETIME:%d$s
```

Die Skriptzeile:

```
stack r_pp_light rej 3 3 -norm=addscale -output_norm -out=$defstack
```

wird den Stack unter folgendem Namen speichern:

```
Result_Rosette_Nebula_2022-01-24_12000s.fit
```

Seit Siril 1.2.0 ist der Standardname für das Stacking als `$sequencename$stacked` (das _-Zeichen wird hinzugefügt, falls nicht vorhanden) definiert. Dies ähnelt dem bisherigen Verhalten früherer Versionen, außer dass jetzt explizit angegeben ist, dass Pfadnamen-Parsing verwendet wird.

22.4 Verwendung von Platzhaltern/Wildcards

Es könnte sein, dass Sie einen Schlüsselwert im Namen Ihres Masterframes verwenden möchten, der nicht mit dem Schlüsselwert in den zu kalibrierenden Bildern übereinstimmt. Anhand eines Beispiels wird es vielleicht etwas deutlicher. Angenommen, Sie möchten im Namen eines Masterflats die Belichtungszeit festhalten. Etwa so:

```
FLAT_1.32s_Halpha_G120_O30_bin1.fit
```

Wenn Sie ein Feld $EXPTIME:%0.2f$ in `$defflat` einfügen, wird es beim Kalibrierungsschritt zu einem Fehler kommen. Einfach deshalb, weil der EXPTIME-Schlüssel aus dem zu kalibrierenden Light gelesen wird und nicht aus dem Flat.

Um diese Situation zu umgehen, können Sie in den zu parsenden Ausdrücken Platzhalter/Wildcards verwenden:

```
FLAT_$*EXPTIME:%0.2f$_$FILTER:%s$_G$GAIN:%d$_O$OFFSET:%d$_bin$XBINNING:%d$
```

Beachten Sie das Symbol *, das direkt vor `EXPTIME` steht.

Dieses Symbol hat folgende Bedeutung:

- Im `Schreibmodus`, also grundsätzlich beim Stacken Ihres Masterflats, wird das Feld `EXPOSURE` zur Bildung des zu speichernden Dateinamens verwendet. Im obigen Beispiel würden Sie dann effektiv unter `FLAT_1.32s_Halpha_G120_O30_bin1.fit` speichern.
- Im `Lesemodus`, also beim Kalibrieren Ihrer Lights, wird das Feld `EXPOSURE` durch * ersetzt. Bei der suche nach einer solchen Datei holt Siril alle Dateien, die dem Muster `FLAT_*_Halpha_G120_O30_bin1.fit` entsprechen. Hoffentlich ist Ihre Namenskonvention robust genug, damit nur genau eine passende Datei gefunden und zur Kalibrierung verwendet wird.

Dies kann auch nützlich sein, um Master-Verzerrungsdateien anzugeben. Sie könnten beispielsweise Folgendes verwenden:

```
~/astro/masters/distos/DISTO_$*DATE-LOC:dm12$_$INSTRUMEN:%s$_$TELESCOP:%s$_bin$XBINNING:
↪%d$.wcs
```

Beachten Sie das Symbol * direkt vor `DATE-LOC`.

Wenn vor einem `DATE`-Header-Schlüssel ein Platzhalter verwendet wird, ist die im Lesemodus zurückgegebene Datei die Datei mit dem nächstgelegenen Datum, das kleiner oder gleich dem der Datei ist, deren Header analysiert wird.

Warnung

Falls Siril mehr als eine Datei im `Lesemodus` findet, wird es eine Warnung in der Konsole ausgeben und die neueste Datei auswählen. Da dies möglicherweise nicht zum gewünschten Ergebnis führt, sollten Sie in diesem Fall Ihre Namenskonvention für Dateien überdenken.

22.5 Umgang mit Duplikaten

In manchen Fällen gibt es mehrere Schlüsselwörter für denselben Wert. Das liegt daran, dass Softwareentwickler dieselben Schlüsselwörter verwenden oder neue schaffen können. Siril versucht daher, Duplikate zu erkennen und zu verwalten. Hier ist eine Tabelle, die bekannte Duplikate zusammenfasst. Wenn eine Datei so genannte „alternative“ Schlüsselwörter enthält, dann speichert Siril den Wert in der „primären“ Version.

Primäres Schlüsselwort	Alternative
MIPS-HI	CWHITE
MIPS-LO	CBLACK
PROGRAM	SWCREATE
IMAGETYP	FRAMETYP
EXPTIME	EXPOSURE
FILTER	FILT-1
FOCALLEN	FOCAL
CENTALT	OBJCTALT
CENTAZ	OBJCTAZ
XBINNING	BINX
YBINNING	BINY
XPIXSZ	XPIXELSZ, PIXSIZE1, PIXSIZEX, XPIXSIZE
YPIXSZ	YPIXELSZ, PIXSIZE2, PIXSIZEY, YPIXSIZE
CCD-TEMP	CCD_TEMP, CCDTEMP, TEMPERAT, CAMTCCD
OFFSET	BLKLEVEL
CVF	EGAIN
FOCPOS	FOCUSPOS
FOCTEMP	FOCUSTEM
STACKCNT	NCOMBINE
SITELAT	SITE-LAT, OBSLAT
SITELONG	SITE-LON, OBSLONG

KAPITEL 23

Live-Stacking

Live Stacking ist eine Technik in der Astrofotografie, die es ermöglicht, eine Reihe von Bildern in Echtzeit zu stacken, um eine höhere Bildqualität zu erzielen. Im Gegensatz zum herkömmlichen Stacking, bei dem mehrere Bilder nach der Aufnahme kombiniert werden, werden beim Live Stacking die Bilder bereits während der Aufnahme kombiniert. Dies ermöglicht eine sofortige Vorschau des endgültigen Bildes und erlaubt dem Astrofotografen, in Echtzeit Anpassungen vorzunehmen, um die Qualität des Endergebnisses zu verbessern.

Siril 1.2.0 enthält diese Funktion, die jedoch im Moment noch **experimentell** ist. Sie erlaubt es, ein Verzeichnis in Echtzeit zu überwachen und die Bilder, die ankommen, zu stacken, sobald sie eintreffen. Das Stapeln von Bildern kann mit und ohne Dark/Bias/Flats durchgeführt werden. Letztere müssen vorher erstellt werden, wenn Sie sie verwenden wollen.

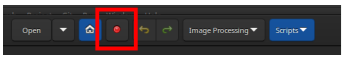
23.1 Live-Stacking (GUI)

Bemerkung

Nur FITS- und RAW-Kamerabilder sind mit dem Livestacking kompatibel.

Um mit dem Livestacking zu beginnen, müssen Sie :

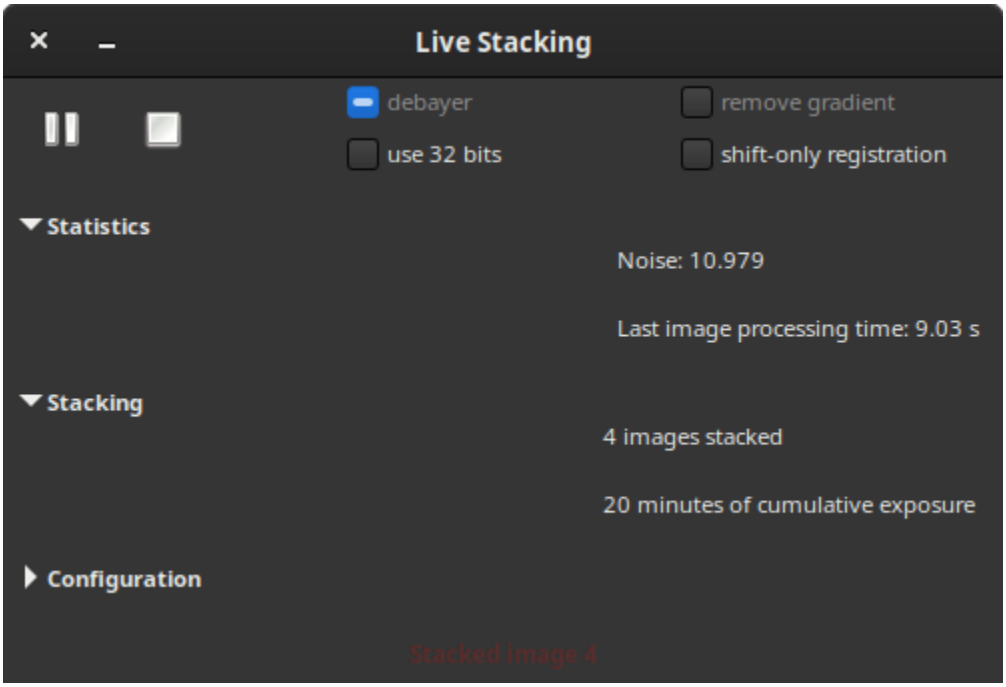
- Ein Arbeitsverzeichnis festlegen, in dem die Fotos nacheinander ankommen sollen.
- Klicken Sie auf die markierte Schaltfläche um eine Live-Stacking-Session zu starten.



Es öffnet sich ein neues Fenster.

Dieses Fenster enthält mehrere Schaltflächen und Optionen. Eine Schaltfläche zum Starten, die beim Anklicken zu einer Schaltfläche zum Pausieren wird, und eine Schaltfläche zum Beenden. Mit der ersten können Sie die Überwachung des Arbeitsverzeichnisses starten oder pausieren, mit der letzten Schaltfläche stoppen Sie das Live-Stacking.

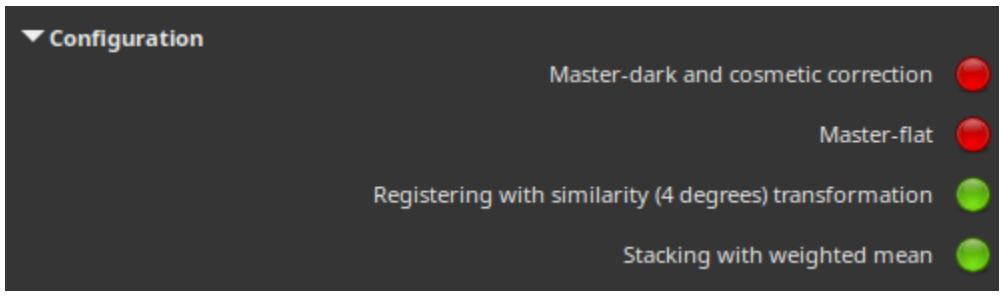
Alle anderen Optionen gehören zum Standard bei der Vorverarbeitung astronomischer Bilder:



- **Debayer**: Die Bayer-Matrix wird in den Dateien erkannt und der Debayer-Algorithmus wird automatisch aktiviert. Dies ist eher eine Information als eine Option.
- **32 Bits benutzen**: Verwenden Sie 32 Bit für die Bildverarbeitung. Dies ist langsamer und in der Regel ohne Vorteile in Bezug auf die Qualität für Live-Stacking.
- **Gradient entfernen**: Entfernt einen linearen Hintergrundgradienten auf den kalibrierten Eingabebildern.
- **Nur-Verschiebung-Registrierung**: Nur Verschiebung von Bildern anstelle von Rotation bei der Registrierung. Dies sollte für Alt-Az-Montierungen oder stark driftende Montierungen deaktiviert werden. Wenn es aktiviert ist, macht es die Verarbeitung der Bilder viel schneller.

Die folgenden 3 Abschnitte enthalten die Informationen, die der Benutzer benötigt, um die Entwicklung des Stacks zu verfolgen.

- **Statistik**: In diesem Abschnitt werden die Entwicklung des Rauschpegels in ADU sowie die Bildverarbeitungszeit angezeigt.
- **Stacking**: In diesem Abschnitt wird die Anzahl der gestackten Bilder und die aufsummierte Gesamtbelichtungszeit angezeigt.
- **Konfiguration** ist ein Abschnitt, der standardmäßig nicht aufgeklappt wird. Nach dem Ausklappen enthält der Rahmen Angaben darüber, ob die Vorverarbeitung mit Hilfe von Masterdateien erfolgt ist, sowie über die Art der Registrierung und des Stackings.



Bemerkung

Um Masterdateien während einer Livestacking-Sitzung zu verwenden, müssen Sie Ihre Kalibrierungsframes zunächst stacken. Danach und vor dem Start der Sitzung laden Sie sie bitte in die Registerkarte *Kalibrierung*. Sie werden dann beim Livestacking berücksichtigt und im *Konfigurations*-Teil des Livestack-Dialogs angezeigt.

23.2 Live-Stacking (ohne GUI/Headless)

Es ist möglich, Live-Stacking von der Kommandozeile aus zu benutzen. Dazu sind nur 3 Befehle notwendig, die im Folgenden erklärt werden.

- Die erste, *start_ls* startet die Livestacking-Sitzung. Es ist möglich, dem Befehl Dark- und Flat-Bilder als Argumente zu übergeben, um die Bilder während des Live-Stackings zu kalibrieren.

  Siril Kommandozeile

  ```
  start_ls [-dark=filename] [-flat=filename] [-rotate] [-32bits]
  ```

  Initialisiert eine Livestacking-Sitzung unter Verwendung der optionalen Kalibrierungsdateien und wartet auf Eingabedateien, die durch den Befehl LIVESTACK bereitgestellt werden, bis STOP_LS aufgerufen wird. Standardmäßig wird die Registrierung nur mit Verschiebung und 16-Bit-Verarbeitung durchgeführt, da dies schneller ist. Dies kann mit **-rotate** und **-32bits** auf Rotation und 32 Bit geändert werden

  Beachten Sie, dass die Live-Stacking-Befehle Siril in einen Zustand versetzen, in dem es nicht in der Lage ist, andere Befehle zu verarbeiten. Nach START_LS können nur noch LIVESTACK, STOP_LS und EXIT aufgerufen werden, bis STOP_LS aufgerufen wird, um Siril in den normalen, nicht-Live-Stacking-Zustand zurückzubringen

  Verweise: *livestack*, *stop_ls*, *exit*

- Der Befehl *livestack* muss auf jedes Bild angewendet werden, das Sie stacken möchten.

  Siril Kommandozeile

  ```
  livestack filename
  ```

  Verarbeitet das bereitgestellte Bild für das Live-Stacking. Nur nach START_LS möglich. Der Prozess umfasst das Kalibrieren der eingehenden Datei, sofern in START_LS konfiguriert, das Debayern, wenn es sich um ein OSC-Image handelt, das Registrieren und Stacken. Das temporäre Ergebnis wird in der Datei live_stack_00001.fit gespeichert, bis eine neue Option zum Ändern hinzugefügt wird

  Verweis: *start_ls*

- Schließlich wird mit dem Befehl *stop_ls* die Live-Stacking-Sitzung beendet.

Siril Kommandozeile

```
stop_ls
```

Beendet die Live-Stacking-Sitzung. Nur möglich nach START_LS

Verweis: *start_ls*

KAPITEL 24

Befehle

Die folgende Seite zeigt alle Befehle die in Siril verfügbar sind 1.4.4.

Sie können durch einen Klick auf dieses Icon auf einen Index zugreifen .

Wenn Befehle mit dem Icon markiert sind, können Sie in Skripten benutzt werden, während Befehle die mit diesem Icon in Skripten nicht möglich sind.

Tipp

For all the sequence commands typed in the command bar of the GUI, you can replace argument **sequencename** with a `.` if the sequence to be processed is already loaded.

Tipp

Wenn Sie ein Argument angeben möchten, das eine Zeichenfolge mit Leerzeichen enthält, beispielsweise einen Dateinamen, müssen Sie das gesamte Argument und nicht nur die Zeichenfolge in Anführungszeichen setzen. Sie sollten also beispielsweise „command „-filename=My File.fits““ verwenden, **nicht** „command -filename="My File.fits"``.

addmax

```
addmax filename
```

Berechnet ein neues Bild durch Kombination des Bildes im Speicher mit dem Bild **filename**. An jeder Pixelposition wird der neue Wert auf das Maximum des aktuellen Bildes und im Bild **filename** bestimmt

asinh

```
asinh [-human] stretch { [offset] [-clipmode=] }
```

Streckt das Bild, um schwache Objekte mithilfe einer hyperbolischen Arcussinus-Transformation anzuzeigen. Das obligatorische Argument **stretch**, normalerweise zwischen 1 und 1000, gibt die Stärke der Streckung an. Der Schwarzpunkt kann verschoben werden, indem ein optionales Argument **offset** im normalisierten Pixelwert von [0, 1] angegeben wird. Schließlich ermöglicht die Option **-human** die Verwendung der Lichtempfindlichkeit des menschlichen Auges zur Berechnung der Leuchtdichte, die zur Berechnung des Streckungswerts für jedes Pixel verwendet wird, anstelle des einfachen Mittelwerts der Kanalpixelwerte. Diese Streckungsmethode bewahrt die Helligkeit des L*a*b*-Farbraums. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend.

autoghs

```
autoghs [-linked] shadowsclip stretchamount [-b=] [-hp=] [-lp=] [-clipmode=]
```

Anwendung der verallgemeinerten hyperbolischen Streckung mit einem Symmetriepunkt SP, definiert als k.sigma vom Median jedes Kanals (der angegebene **shadowclip**-Wert ist hier k und kann negativ sein). Standardmäßig werden SP und die Streckung pro Kanal getrennt berechnet; durch Angabe von **-linked** kann die Streckung als Mittelwert der Kanäle berechnet werden. Der Stretchfaktor **D** wird als zweites, obligatorisches Argument angegeben.

Implizite Werte von 13 für **B**, wodurch es sehr auf den SP-Helligkeitsbereich fokussiert ist, 0,7 für **HP** und 0 für **LP** werden verwendet, können aber mit den gleichnamigen Optionen geändert werden. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend

Automatisches Strecken

```
autostretch [-linked] [shadowsclip [targetbg]]
```

Streckt das aktuell geladene Bild mit unterschiedlichen Parametern für jeden Kanal (nicht verknüpft), sofern nicht **-linked** übergeben wird. Argumente sind optional, **shadowclip** ist der Schatten-Beschneidungspunkt, gemessen in Sigma-Einheiten von der Spitze des Haupthistogramms (Der Defaultwert ist -2,8), **targetbg** ist der Ziel-Hintergrundwert, der dem Bild seine endgültige Helligkeit gibt. Bereich [0, 1], Defaultwert ist 0,25. Die Defaultwerde werden in der Auto-Stretch-Funktion der grafischen Benutzeroberfläche benutzt.

Verwenden Sie nach der Farbkalibrierung nicht die nicht verknüpfte Version, da sie den Weißabgleich verändert

Hintergrund

```
bg
```

Gibt den Hintergrund-Wert des aktuell geladenen Bildes aus

bgnoise

```
bgnoise
```

Gibt den Rauschwert des aktuell geladenen Bildes aus

Für weitere Informationen, siehe *Dokumentation Statistik*

Binning (Numerischer Wert)

```
binxy coefficient [-sum]
```

Berechnet das numerische Binning des geladenen Bildes (Summe der Pixel 2x2, 3x3..., ähnlich dem analogen Binning einer CCD-Kamera). Wenn das optionale Argument **-sum** übergeben wird, wird die Summe der Pixel berechnet, ansonsten der Durchschnitt der Pixel

Auswahl Rechteck

```
boxselect [-clear] [x y width height]
```

Erzeugt einen Auswahlbereich im aktuell geladenen Bild mit den Argumenten **x**, **y**, **Breite** und **Höhe**, wobei **x** und **y** die Koordinaten der linken oberen Ecke, beginnend bei (0, 0), und **width** und **height** die Größe der Auswahl sind. Das Argument **-clear** löscht jeden Auswahlbereich. Wenn kein Argument übergeben wird, wird die aktuelle Auswahl ausgegeben

calibrate

```
calibrate sequencename [-bias=filename] [-dark=filename] [-flat=filename] [-cc=dark␣
↪[siglo sighi] || -cc=bpm bpmfile] [-cfa] [-debayer] [-fix_xtrans] [-equalize_cfa] [-
↪opt[=exp]] [-all] [-prefix=] [-fitseq]
```

Kalibrierung der Sequenz **Sequenzname** unter Verwendung der im Argument angegebenen bias-, dark- und flat-Dateien.

Für Bias kann anstelle eines Bildes ein einheitlicher Wert angegeben werden, indem ein mit einem =-Zeichen beginnender Anführungszeichenausdruck eingegeben wird, z. B. -bias="=256" oder -bias="=64*$OFFSET" (synthetischer Bias).

Standardmäßig ist die kosmetische Korrektur nicht aktiviert. Wenn Sie eine solche Korrektur wünschen, müssen Sie sie mit der Option **-cc=** angeben.

Sie können **-cc=dark** verwenden, um Hot- und Coldpixel aus dem Masterdark zu erkennen (ein Masterdark muss mit der Option **-dark=** angegeben werden), optional gefolgt von **siglo** und **sighi** für Cold- bzw. Hotpixel. Ein Wert von 0 deaktiviert die Korrektur. Wenn keine Sigmawerte angegeben werden, wird nur die Erkennung von Hotpixeln mit einem sigma von 3 angewendet.

Alternativ können Sie **-cc=bpm** gefolgt von dem Pfad zu Ihrer Bad Pixel Map verwenden, um anzugeben, welche Pixel korrigiert werden müssen. Eine Beispieldatei kann mit einem *find_hot*-Befehl auf einem Masterdark erhalten werden.

Drei Optionen gelten für Farbbilder (im CFA-Format): **-cfa** für kosmetische Korrekturen, **-debayer**, um Bilder vor dem Speichern zu entmosaikisieren, und **-equalize_cfa**, um die mittlere Intensität der RGB-Ebenen des Master-Flats auszugleichen, um eine Färbung des kalibrierten Bildes zu vermeiden.

Die Option **-fix_xtrans** widmet sich X-Trans-Dateien, indem sie eine Korrektur auf Dark- und Biasframes anwendet, um ein recheckiges Muster zu entfernen welches durch den Autofokus verursacht wird.

Es ist auch möglich, die Dunkelbildsubtraktion mit **-opt** zu optimieren, was die Bereitstellung von Bias und Dunkelmastern erfordert und automatisch den auf die Darks anzuwendenden Koeffizienten berechnet, oder den Koeffizienten dank des Belichtungs-Schlüsselworts **-opt berechnet =exp** berechnet.

Standardmäßig werden die als ausgeschlossen markierten Bilder nicht verarbeitet. Das Argument **-all** kann verwendet werden, um die Verarbeitung aller Bilder zu erzwingen, auch wenn sie als ausgeschlossen markiert sind.

Der Name der Ausgabesequenz beginnt mit dem Präfix "pp_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde.

Wenn **-fitseq** angegeben wird, ist die Ausgabesequenz eine FITS-Sequenz (einzelne Datei)

calibrate_single

```
calibrate_single imagename [-bias=filename] [-dark=filename] [-flat=filename] [-cc=dark␣
↪[siglo sighi] || -cc=bpm bpmfile] [-cfa] [-debayer] [-fix_xtrans] [-equalize_cfa] [-
↪opt[=exp]] [-prefix=]
```

Kalibriert das Bild **imagename** unter Verwendung der im Argument angegebenen Bias-, Dark- und Flat-Dateien.

Für Bias kann anstelle eines Bildes ein einheitlicher Wert angegeben werden, indem ein mit einem =-Zeichen beginnender Anführungszeichenausdruck eingegeben wird, z. B. -bias="=256" oder -bias="=64*$OFFSET" (synthetischer Bias).

Standardmäßig ist die kosmetische Korrektur nicht aktiviert. Wenn Sie eine solche Korrektur wünschen, müssen Sie sie mit der Option **-cc=** angeben.

Sie können **-cc=dark** verwenden, um Hot- und Coldpixel aus dem Masterdark zu erkennen (ein Masterdark muss mit der Option **-dark=** angegeben werden), optional gefolgt von **siglo** und **sighi** für Cold- bzw. Hotpixel. Ein Wert von 0 deaktiviert die Korrektur. Wenn keine Sigmawerte angegeben werden, wird nur die Erkennung von Hotpixeln mit einem sigma von 3 angewendet.

Alternativ können Sie **-cc=bpm** gefolgt von dem Pfad zu Ihrer Bad Pixel Map verwenden, um anzugeben, welche Pixel korrigiert werden müssen. Eine Beispieldatei kann mit einem *find_hot*-Befehl auf einem Masterdark erhalten werden.

Drei Optionen gelten für Farbbilder (im CFA-Format): **-cfa** für kosmetische Korrekturen, **-debayer**, um Bilder vor dem Speichern zu entmosaikisieren, und **-equalize_cfa**, um die mittlere Intensität der RGB-Ebenen des Master-Flats auszugleichen, um eine Färbung des kalibrierten Bildes zu vermeiden.

Die Option **-fix_xtrans** widmet sich X-Trans-Dateien, indem sie eine Korrektur auf Dark- und Biasframes anwendet, um ein recheckiges Muster zu entfernen welches durch den Autofokus verursacht wird.

Es ist auch möglich, die Dunkelbildsubtraktion mit **-opt** zu optimieren, was die Bereitstellung von Bias und Dunkelmastern erfordert und automatisch den auf die Darks anzuwendenden Koeffizienten berechnet, oder den Koeffizienten dank des Belichtungs-Schlüsselworts **-opt berechnet =exp** berechnet

Der Ausgabedateiname beginnt mit dem Präfix "pp_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

Fähigkeiten

```
capabilities
```

Auflistung der Siril-Fähigkeiten, basierend auf den Kompilierungsoptionen und Laufzeit

catsearch

```
catsearch name
```

Sucht ein Objekt nach **Name** und fügt es dem Benutzerkatalog für Anmerkungen hinzu. Das Objekt wird zunächst in den Beschriftungskatalogen gesucht, wenn es nicht gefunden wird, wird eine Anfrage an SIMBAD gestellt.

Das Objekt kann ein Objekt des Sonnensystems sein. In diesem Fall ist vor dem Objektnamen ein Präfix "a:" für Asteroid, "p:" für Planet, "c:" für Komet, "dp:" für Zwergplanet oder "s:" für natürlicher Satellit erforderlich. Die Suche erfolgt nach dem Datum, der Uhrzeit und dem Beobachtungsort, die in der Kopfzeile des Bildes angegeben

sind, unter Verwendung des "IMCCE Miriade Service
<https://ssp.imcce.fr/webservices/miriade/howto/ephemcc/#howto-sso>"

Farbkonvertierungsmatrix

```
ccm m00 m01 m02 m10 m11 m12 m20 m21 m22 [gamma]
```

Wendet eine Farbkonvertierungsmatrix auf das aktuelle Bild an.

Es gibt 9 obligatorische Argumente, die den 9 Matrixelementen entsprechen:

m00, m01, m02
m10, m11, m12
m20, m21, m22

Ein zusätzliches zehntes Argument **[gamma]** kann bereitgestellt werden: Wenn es weggelassen wird, wird standardmäßig 1,0 verwendet.

Diese werden nach folgenden Formeln auf jedes Pixel angewendet:

r' = (m00 * r + m01 * g + m02 * b)^(-1/gamma)
g' = (m10 * r + m11 * g + m12 * b)^(-1/gamma)
b' = (m20 * r + m21 * g + m22 * b)^(-1/gamma)

cd

```
cd directory
```

Setzt ein neues aktuelles Arbeitsverzeichnis.

Das Argument **directory** kann das Tokten ~ enthalten, welches zum Pfad des Heimatverzeichnisses des aktuellen Benutzers erweitert wird. Verzeichnisse mit Leerzeichen im Namen können durch einfache oder doppelte Anführungszeichen geschützt werden

cdg

```
cdg
```

Ermittelt die Koordinaten des Schwerpunkts des Bildes. Nur Pixel mit Werten über 15,7 % der maximalen ADU und vier Nachbarn, die dieselbe Bedingung erfüllen, werden zur Berechnung herangezogen, und er wird nur berechnet, wenn es mindestens 50 davon gibt

clahe

```
clahe cliplimit tileSize
```

Gleicht das Histogramm eines Bildes mit Hilfe der Kontrastbegrenzten Adaptiven Histogrammangleichung (CLAHE) aus.

cliplimit legt den Schwellenwert für die Kontrastbegrenzung fest.
tilesize legt die Größe des Rasters für die Histogramm-Angleichung fest. Das Eingabebild wird in rechteckige Kacheln gleicher Größe unterteilt

clear

```
clear
```

Leert die Konsolenausgabe in der Benutzeroberfläche

clearstar

```
clearstar
```

Entfernt alle Sterne die im RAM gespeichert und auf dem Bildschirm angezeigt sind

close

```
close
```

Schließt das geöffnete Bild und die geöffnete Sequenz ordnungsgemäß, falls vorhanden

conesearch

```
conesearch [limit_magnitude] [-cat=] [-phot] [-obscode=] [-tag={on|off}] [-log={on|off}]␣
↪[-trix=] [-out=]
```

Zeigt standardmäßig Sterne aus dem lokalen Katalog für das geladene, astrometrisch gelöste Bild an, bis zur angegebenen Grenzgröße (**limit_magnitude**) (standardmäßig 13 für die meisten Kataloge, außer 14.5 für aavso_chart, 20 für solsys und ohne Grenzgröße für pgc).

Ein alternative Onlinekatalog kann durch Angabe von **-cat=** angegeben werden mit folgenden Werten

- for stars: tycho2, nomad, gaia, localgaia, ppmxl, bsc, apass, gcvs, vsx, simbad, aavso_chart

- for exoplanets: exo

- for deep-sky: pgc

-für Objekte des Sonnensystems: solsys (der nächstgelegene`IAU observatory code <https://vo.imcce.fr/webservices/data/displayIAUObsCodes.php>`__ kann mit dem Argument **-obscode=** für eine bessere Positionsgenauigkeit übergeben werden)

Für Sternkataloge die photometrische Daten enthalten, werden Sterhen ohne B-V-Informationen beibehalten; sie können durch Angabe von **-phot** ausgeschlossen werden

Das Argument **-trix=** kann anstelle eines Katalogs gefolgt von einer Zahl zwischen 0 und 511 übergeben werden, um Sterne zu zeichnen, die im lokalen Katalog-Trixel der Ebene 3 enthalten sind (hauptsächlich für die Verwendung durch Entwickler)

Bei einigen Katalogen (bsc, gcvs, pgc, exo, aavso_chart und solsys) werden standardmäßig auch die Namen neben den Markern in der Anzeige (nur GUI) angezeigt und im Protokoll aufgeführt. Bei anderen Katalogen mit einer größeren Anzahl von Objekten, namentlich vsx und simbad, können diese Informationen ebenfalls angezeigt werden, aber da sie die Anzeige überladen könnten, sind sie standardmäßig nicht aktiviert. Dieses Verhalten kann mit den Optionen **-tag=on|off** zur Anzeige der Namen neben den Markern und **-log=on|off** zur Auflistung der Objekte im Konsolenprotokoll ein- und ausgeschaltet werden

Die Liste der im Bild vorhandenen Elemente kann optional durch Übergabe des Arguments **-out=** in einer CSV-Datei gespeichert werden

convert

```
convert basename [-debayer] [-fitseq] [-ser] [-start=index] [-out=]
```

Konvertiert alle Bilder des aktuellen Arbeitsverzeichnisses, die in einem unterstützten Format vorliegen, in eine Siril-Sequenz von FITS-Bildern (mehrere Dateien) oder eine FITS-Sequenz (einzelne Datei), wenn **-fitseq** angegeben wird, oder eine SER-Sequenz (einzelne Datei), wenn **-ser** angegeben wird. Das Argument **basename** ist der Basisname der neuen Sequenz, Zahlen und die Erweiterung werden dahinter gesetzt.

Bei FITS-Bildern versucht Siril, einen symbolischen Link zu erstellen; ist dies nicht möglich, werden die Dateien kopiert. Die Option **-debayer** wendet Debayering auf CFA(Farb-)-Eingabebilder an; in diesem Fall wird kein symbolischer Link erstellt.

-start=index setzt die Startindexnummer, nützlich um eine bestehende Sequenz fortzusetzen (wird nicht mit -fitseq oder **-ser** verwendet; stellen Sie sicher, dass Sie die Ziel-.seq entfernen oder löschen, falls sie in diesem Fall existiert).

Die Option **-out=** ändert das Ausgabeverzeichnis zum angegebenen Argument.

Siehe auch CONVERTRAW und LINK

Verweise: *convertraw*, *link*

convertraw

```
convertraw basename [-debayer] [-fitseq] [-ser] [-start=index] [-out=]
```

Identisch mit CONVERT, konvertiert jedoch nur DSLR-RAW-Dateien, die im aktuellen Arbeitsverzeichnis gefunden werden

Verweise: *convert*

cosme

```
cosme [filename].lst
```

Wendet den lokalen Mittelwert auf eine Gruppe von Pixeln des gespeicherten Bildes an (kosmetische Korrektur). Die Koordinaten dieser Pixel befinden sich in einer Text-Datei [.lst-Datei]. Der Befehl FIND_HOT kann auch für einzelne Hotpixel verwendet werden, aber es ist eine manuelle Bedienung erforderlich, um Zeilen oder Spalten zu entfernen. COSME ist in der Lage, nach der Kalibrierung verbleibende heiße und kalte Pixel zu korrigieren.

Anstatt die Liste der schlechten Pixel zu liefern, ist es auch möglich, sie im aktuellen Bild mit dem Befehl FIND_COSME aufzuspüren

Verweise: *find_hot*, *find_cosme*

Dateiformat für die Liste der schlechten Pixel: * Zeilen der Form "P x y" korrigieren das Pixel an den Koordinaten (x, y). type ist ein optionales Zeichen (C oder H), das Siril angibt, ob das aktuelle Pixel kalt oder heiß ist. Diese Zeile wird durch den Befehl FIND_HOT erzeugt, aber Sie können die beiden folgenden Zeilentypen auch manuell hinzufügen *

Zeilen der Form *C x 0* korrigieren die schlechte Spalte an den Koordinaten x. * Zeilen der Form *L y 0* korrigieren die schlechte Zeile an den Koordinaten y.

cosme_cfa

```
cosme_cfa [filename].lst
```

Gleiche Funktion wie COSME, jedoch für RAW-CFA-Bilder

Verweis: *cosme*

crop

```
crop [x y width height]
```

Beschneidet einen ausgewählten Bereich des geladenen Bildes.

Falls eine Auswahl aktiv ist, sind keine weiteren Argumente erforderlich. Andernfalls oder in Skripten müssen Argumente angegeben werden, wobei **x** und **y** die Koordinaten der linken oberen Ecke und **width** und *height** die größe der Auswahl sind. Alternativ kann die Auswahl auch mit dem BOXSELECT-Befehl im geladenen Bild getroffen werden

Links: *boxselect*

ddp

```
ddp level coef sigma
```

Führt eine DDP (Digital Development Processing) durch, wie sie zuerst von Kunihiko Okano beschrieben wurde. Diese Implementierung ist diejenige, die in IRIS beschrieben wird.

Sie kombiniert eine lineare Verteilung auf niedrige Niveaus (unter **level**) and eine nicht-lineare Verteilung auf hohe Niveaus.
Es verwendet einen Gauß-Filter mit der Standardabweichung **sigma** und multipliziert das resultierende Bild mit **coef**. Typische Werte für **sigma** liegen zwischen 0,7 und 2. Das Argument level sollte bei 16-Bit-Bildern im Bereich [0,

65535] liegen und kann bei 32-Bit-Bildern entweder im Bereich [0, 1] oder [0, 65535] angegeben werden; in diesem Fall wird es automatisch skaliert

denoise

```
denoise [-nocosmetic] [-mod=m] [ -vst | -da3d | -sos=n [-rho=r] ] [-indep]
```

Entrauscht dsa Bild mit dem von *Lebrun, Buades und Morel <https://www.ipol.im/pub/art/2013/16>* beschriebenen Algorithmus.

Es wird dringend empfohlen, vor der Entrauschung eine kosmetische Korrektur auszuführen, um Salz- und Pfefferrauschen zu entfernen. Wenn die jedoch bereits zu einem früheren Zeitpunkt im Arbeitsablauf erfolgt ist, kann es hier mit dem optionalen Befehl **-nocosmetic** deaktiviert werden.

Ein optionales Argument **-mod=m** kann angegeben werden, wobei 0 <= m <= 1. Das Ausgabepixel wird dann *out=m x d + (1 - m) x in* berechnet, wobei *d* der entrauschte Pixelwert ist. Bei einem Modulationswert von 1 wird keine Modulation angewendet. Wird der Parameter nicht angegeben, wird er standardmäßig auf 1 gesetzt.

Das optionale argument **-vst** kann verwendet werden, um die verallgemeinerte varianzstabilisierende Anscombe-Transformation (Generalised Anscombe Variance Stabilizing Transform) vor NL-Bayer anzuwenden. Dies ist nützlich bei photonenarmen Bildern, wie z.B. Einzelbildern, bei denen das Rauschen einer Poisson- oder Poisson-Gauß-Verteilung folgt und nicht primär gaußförmig ist. Es kann nicht in Verbindung mit DA3D oder SOS verwendet werden, und für die Entrauschung von gestackten Bildern ist es in der Regel nicht von Vorteil.

Mit dem ptionalen Argument **-da3d** kann die datenadaptive Dual-Domain-Entrauschung (DA3D) als letzter Entrauschungsalgorithmus aktiviert werden. Dabei wird die Ausgabe von BM3D als Vorgabe zur Verfeinerung der Rauschunterdrückung verwendet. Das verbessert die Detailwiedergabe und reduziert Treppenartefakte.

Das optionale Argument **-sio=n** kann verwendet werden, um das iterative Denoise-Boosting nach dem SOS-Prinzip (Strengthen-Operate-Subtract) zu aktivieren, wobei die Anzahl der Iterationen durch n festgelegt wird. Dieser Booster kann insbesondere dann bessere Ergebnisse liefern, wenn der NL-Bayes-Algorithmus ohne Booster Artefakte im Hintergrundbereich erzeugt. Wenn sowohl -da3d als auch -sos=n angegeben sind, gilt die zuletzt angegebene Option.

Das optionale Argument **-rho=r** kann angegeben werden, wobei 0 < r < 1. Dies wird vom SOS-Booster verwendet, um den Anteil des verrauschten Bildes zu bestimmen, der dem Zwischenergebnis zwischen den einzelnen Iterationen hinzugefügt wird. Wenn -sos=n nicht angegeben wird, wird der Parameter ignoriert.

Standardmäßig werden DA3D oder SOS nicht angewendet, da die Verbesserung der Rauschunterdrückung in der Regel relativ gering ist und diese Techniken zusätzliche Verarbeitungszeit erfordern.

In sehr seltenen Fällen können bei der Entrauschung von Farbbildern blockartige Farbartefakte in der Ausgabe entstehen. Das optionale Argument **-indep** kann verwendet werden, um dies zu verhindern, indem jeder Kanal separat entrauscht wird. Dies ist zwar langsamer, beseitigt aber die Artefakte

dir

```
dir
```

Listet Dateien und Verzeichnisse im Arbeitsverzeichnis auf

Dieser Befehl ist nur unter Microsoft Windows verfügbar, für den entsprechenden Befehl unter Linux und MacOS siehe *ls*.

disto

```
disto [clear]
```

Zeigt das Verzerrungsfeld eines astrometrisch gelösten Bildes mit Verzerrungstermen an

Option **clear** angeben zum deaktivieren

dumpheader

```
dumpheader
```

Zeigt den FITS-Header-Wert für den angegebenen Schlüssel des geladenen Bildes in der Konsole an

entropy

```
entropy
```

Berechnet die Entropie des geladenen Bildes für den dargestellten Layer, aber nur im Auswahlbereich - sofern einer vorhanden ist - ansonsten im Gesamtbild. Entropie ist eine Möglichkeit, den Rauschpegel oder den Detail-Grad eines Bildes zu bestimmen

epf

```
epf [-guided] [-d=] [-si=] [-ss=] [-mod=] [-guideimage=]
```

Wendet einen kantenerhaltenden Filter an. Standardmäßig wird ein bilateraler Filter angewendet; ein geführter Filter kann mit dem Argument **-guided** angegeben werden. Der Filterdurchmesser ist standardmäßig auf 3 eingestellt und kann mit **-d=** festgelegt werden. Seien Sie vorsichtig mit d-Werten größer als 20, da der Algorithmus rechenintensiv sein kann.

Der Sigma-Wert für die Intensitätsfilterung kann mit **-si=** und der räumliche Sigma-Wert mit **-ss=** eingestellt werden. Sigma-Werte stellen die Differenz der Pixelwerte dar, über die der Filter stark wirkt: Für 32-Bit-Bilder sollte der Wert zwischen 0 und 1,0 liegen, während er für 16-Bit-Bilder zwischen 0 und 65535 liegen sollte. Wenn nichts angegeben wird, sind beide Werte standardmäßig auf 11 gesetzt. Wenn **-d=0** gesetzt ist, wird der Filterdurchmesser automatisch basierend auf dem Wert von **-ss** gesetzt. *Beachten Sie, dass beim Anwenden eines geführten Filters nur * *-sc** gilt.*

Bei der Angabe eines geführten Filters kann mit **-guideimage=** ein Führungsbild festgelegt werden. Wenn kein Führungsbild angegeben wird, wird standardmäßig ein selbstgeführter Filter ausgeführt. *Hinweis: Das Führungsbild muss dieselben Abmessungen haben wie das zu filternde Bild!*

Die Stärke des Filters kann mit dem Argument **-mod=** moduliert werden. Bei mod = 1,0 wird die volle Wirkung des Filters angewendet; bei mod kleiner als 1,0 wird ein Teil des Originalbilds mit dem Ergebnis gemischt und bei mod = 0,0 wird keine Filterung angewendet.

exit

```
exit
```

Beendet das Programm

extract

```
extract NbPlans
```

Extrahiert **NbPlans** Ebenen der Wavelet-Domäne des geladenen Bildes
Siehe auch WAVELET und WRECONS. Für Farbextration, eiehe SPLIT

Verweise: *wavelet*, *wrecons*, *split*

extract_Green

```
extract_Green
```

Extrahiert das grüne Signal aus dem geladenen CFA-Bild. Es liest die Bayer-Matrix-Informationen aus dem Bild oder den Voreinstellungen und exportiert nur die gemittelten Grünfilterdaten als neue, halbgroße FITS-Datei. Es wird eine neue Datei erstellt, deren Name mit dem Präfix "Green_" beginnt

extract_Ha

```
extract_Ha [-upscale]
```

Extrahiert das H-Alpha-Signal aus dem geladenen CFA-Bild. Es liest die Bayer-Matrix-Informationen aus dem Bild oder den Voreinstellungen und exportiert nur die Rotfilterdaten als eine neue, halb so große FITS-Datei. Wenn das Argument **-upscale** angegeben wird, wird die Ausgabe x2 hochskaliert, um die volle Sensorauflösung zu erreichen, z.B. um mit anderen Bildern der gleichen Sensorfamilie übereinzustimmen. Es wird eine neue Datei erstellt, deren Name mit dem Präfix "Ha_" versehen ist

extract_HaOIII

```
extract_HaOIII [-resample=]
```

Extrahiert das H-Alpha und O-III-Signal aus dem geladenen CFA-Bild. Es liest die Bayer-Matrix-Informationen aus dem Bild oder den Voreinstellungen und exportiert nur die Rotfilterdaten als H-Alpha als neue, halbgroße FITS-Datei (Wie EXTRACTHA) und behält die beiden Anderen für OIII mit einem interpolierten Wert für die roten Pixel . Es werden neue Dateien erstellt, deren Name mit dem Präfix "Ha_" und "OIII_" beginnt

Das optionale Argument **-resample={ha|oiii}** legt fest, ob das Ha-Bild hoch- oder das OIII-Bild herunterskaliert werden soll um Bilder mit der selben Größe zu erhalten. Wenn dieses Argument nicht angegeben wird, wird kein Resampling durchgeführt und das OIII-Bild hat die doppelte Höhe und Breite des Ha-Bildes

fdiv

```
fdiv filename scalar
```

Dividiert das geladene Bild durch das im Argument angegebene Bild. Das resultierende Bild wird mit dem Wert des Arguments **scalar** multipliziert. Siehe auch IDIV

Verweis: *idiv*

ffill

```
ffill value [x y width height]
```

Gleicher Befehl wie "FILL", aber es wird symmetrisch in einem mit der Maus oder dem Befehl BOXSELECT definierten Bereich ausgeführt. Findet Verwendung im Bereich der FFT (Fast Fourier Transformation)

Verweise: *fill*, *boxselect*

fftd

```
fftd modulus phase
```

Wendet eine Fast-Fourier-Transformation auf das geladene Bild an. **modulus** und **phase**, die im Argument angegeben sind, sind die Dateinamen der gespeicherten FITS-Dateien

ffti

```
ffti modulus phase
```

Ermittelt ein korrigiertes Bild unter Anwendung einer inversen Transformation. Als **nodulus** und **phase** werden die Dateien verwendet, die im Argument angegeben werden. Das Ergebnis wird als neues geladenes Bild angezeigt

fill

```
fill value [x y width height]
```

Füllt das gesamte geladene Bild (oder die aktuelle Auswahl) mit Pixeln, die den **Wert** Intensität, ausgedrückt in ADU, haben

find_cosme

```
find_cosme cold_sigma hot_sigma
```

Führt eine automatische Erkennung heißer/kalter Pixel anhand der angegebenen Schwellwerte (ins Sigma-Einheiten) durch

find_cosme_cfa

```
find_cosme_cfa cold_sigma hot_sigma
```

Gleicher Befehl wie FIND_COSME, aber für CFA-Bilder

Verweis: *find_cosme*

find_hot

```
find_hot filename cold_sigma hot_sigma
```

Speichert eine Listendatei **filename** (Textformat) im Arbeitsverzeichnis, die die Koordinaten der Pixel enthält, die eine Intensität **hot_sigma** mal höher und **cold_sigma** mal niedriger als die vom geladenen Bild extrahierte Standardabweichung haben. Wir benutzten diesen Befehl generell bei einer Master-Dark-Datei. Der Befehl COSME kann diese Liste fehlerhafter Pixel auf ein geladenes Bild anwenden, siehe auch SEQCOSME, um sie auf eine Sequenz anzuwenden

Verweise: *cosme*, *seqcosme*

Die Zeile `P x y type` korrigiert das Pixel an den Koordinaten (x, y) type ist ein optionales Zeichen (C oder H), das Siril angibt, ob das aktuelle Pixel kalt oder heiß ist. Diese Zeile wird durch den Befehl FIND_HOT erzeugt, aber Sie können auch manuell Zeilen hinzufügen:

Die Zeile `C x 0 type` behebt die fehlerhafte Spalte an den Koordinaten x.

Die Zeile `L y 0 type` behebt die fehlerhafte Linie an den Koordinaten y.

findcompstars

```
findcompstars star_name [-narrow|-wide] [-catalog={nomad|apass}] [-dvmag=3] [-dbv=0.5] [-
↪emag=0.03] [-out=nina_file.csv]
```

Findet automatisch Vergleichssterne im Feld des geladenen Bildes für die photometrische Analyse der Lichtkurve eines Sterne gemäß
- the provided name of the star
- the field of view of the image, reduced to a diameter of its height if **-narrow** is passed, avoiding stars in the corners
- the chosen catalog (APASS by default), can be changed with **-catalog={NOMAD|APASS}**
- the difference in visual magnitude from the variable star, in the range [0, 6] with a default of 3, changed with **-dvmag=**
- the difference in color with the variable star, in the range [0.0, 0.7] of their B-V indices with a default of 0.5, changed with **-dbv=**
- the maximum allowed error on Vmag in the range [0.0, 0.1] with a default of 0.03, changed with **-emag=**.

Die Liste kann optional als CSV-Datei gespeichert werden, die mit der NINA-Vergleichssternliste kompatibel ist, indem der Dateiname mit **-out=** angegeben wird. Ist der angegebene Name der spezielle Wert **auto**, wird er unter Verwendung der Eingabeparameter erzeugt

Siehe auch LIGHT_CURVE

Verweise: *light_curve*

findstar

```
findstar [-out=] [-layer=] [-maxstars=]
```

Erkennt Sterne im aktuell geladenen Bild, die einen höheren Pegel als den von Siril berechneten Schwellenwert haben.
Abschließend wird eine PSF angewendet, und Siril weist alle erkannten Strukturen zurück, die eine Reihe von vorgegebenen Erkennungskriterien nicht erfüllen, die mit dem Befehl SETFINDSTAR eingestellt werden können.
Schließlich wird eine Ellipse um die erkannten Sterne gezogen.

Der optionale Parameter **-out=** ermöglicht es, die Ergebnisse unter dem angegebenen Pfad zu speichern.
Die Option **-layer=** gibt die Ebene an, auf der die Erkennung durchgeführt wird (nur für Farbbilder).
Sie können auch die maximale Anzahl der erkannten Sterne begrenzen, indem Sie der Option **-maxstars=** einen Wert übergeben.

Siehe auch den Befehl CLEARSTAR

Verweise: *psf*, *setfindstar*, *clearstar*

fix_xtrans

```
fix_xtrans
```

Behebt die Fujifilm X-Trans Autofokus-Pixel. im geladenen Bild.

Aufgrund des Phasenerkennungs-Autofokussystems erhalten die für den Autofokus verwendeten Sensorzellen etwas weniger Licht als die umliegenden Sensorzellen. Die Kamera gleicht dies aus und erhöht die Werte dieser speziellen Sensorzellen, was zu einem sichtbaren Quadrat in der Mitte der dunklen/Bias-Bilder führt

fixbanding

```
fixbanding amount sigma [-vertical]
```

Versucht das horizontale oder vertikale Banding im geladenen Bild zu entfernen.
Mit dem Argument **amount** wird die Höhe der Korrektur zwischen 0 und 4 festgelegt.
sigma definiert das Hightlight-Schutzniveau des Algorithmus, wobei ein höheres Sigma einen höheren Schutz bietet, der zwischen 0 und 5 liegt. Werte von 1 und 1 sind oft gut genug.
Die Option **-vertical** ermöglicht das Entfernen von vertikalem Banding, die Standardeinstellung ist horizontal

fmedian

```
fmedian ksize modulation
```

Führt einen Medianfilter der Größe **ksize** x **ksize** (**ksize** MUSS ungerade sein) auf das geladene Bild mit einem Modulationsparameter **modulation** aus.

Das Ausgabepixel wird wie folgt berechnet: out=mod x m + (1 - mod) x in, wobei m der Median-gefilterte Pixelwert ist. Bei einem Modulationswert von 1 wird keine Modulation angewendet

fmul

```
fmul scalar
```

Multipliziert das geladene Bild mit dem im Argument **scalar** angegebenen Wert

gauss

```
gauss sigma
```

Führt einen Gauß-Filter mit dem angegebenen **sigma** auf dem geladenen Bild aus.

Siehe auch UNSHARP, dasselbe mit einem Blendingparameter

Verweise: *unsharp*

get

```
get { -a | -A | variable }
```

Holt einen Wert aus den Einstellungen anhand seines Namens oder listet alle mit **-a** (Liste mit Namen und Werten) oder mit **-A** (detaillierte Liste) auf

Siehe auch SET um Werte zu ändern

Verweise: *set*

getref

```
getref sequencename
```

Gibt Informationen über das Referenzbild der im Argument angegebenen Sequenz aus. Das erste Bild hat den Index 0

ght

```
ght -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -sat]␣
↪[channels]
```

Verallgemeinerte hyperbolische Streckung (Generalized Hyperbolic Stretch) auf der Grundlage der Arbeit des Teams von ghsastro.co.uk.

Das Argument **-D=** bestimmt die Stärke der Streckung, die zwischen 0 und 10 liegt. Dies ist das einzige obligatorische Argument. Mit den folgenden optionalen Argumenten kann die Streckung weiter angepasst werden:

B bestimmt die Intensität der Streckung in der Nähe des Symmetriepunktes, zwischen -5 und 15;

LP definiert einen schattenerhaltenden Bereich zwischen 0 und SP, in dem die Streckung linear erfolgt und die Schattendetails erhalten bleiben;

SP definiert den Symmetriepunkt der Streckung, der zwischen 0 und 1 liegt und an dem die Streckung am stärksten ist;

HP definiert einen Bereich zwischen HP und 1, in dem die Streckung linear erfolgt, wodurch die Details der Glanzlichter erhalten bleiben und eine Aufblähung der Sterne verhindert wird.

Wenn B, LP und SP weggelassen werden, ist der Standardwert 0,0 und bei HP ist der Standardwert 1,0.

Ein optionales Argument (entweder **-human**, **-even** oder **-independent**), kann übergeben werden, um entweder menschlich-gewichtete oder gleichmäßig gewichtete Luminanz oder unabhängige Farbkanäle für Farbstreckungen auszuwählen. Das Argument wird bei Monobildern ignoriert. lternativ gibt das Argument **-sat** an, dass die Streckung anhand der Bildsättigung erfolgt - das Bild muss farbig sein und alle Kanäle müssen ausgewählt sein, damit dies funktioniert.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: Dies können R, G, B, RG, RB oder GB sein. Der Standardwert sind alle Kanäle. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte wie **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend.

grey_flat

```
grey_flat
```

Gleicht die mittlere Intensität der RGB-Ebenen im geladenen CFA-Bild aus. Dies ist derselbe Vorgang, der bei der Kalibrierung bei Flats angewendet wird, wenn die Option „CFA ausgleichen“ verwendet wird

help

```
help [command]
```

Auflistung der verfügbaren Befehle oder Hilfe zu einem Befehl

histo

```
histo channel (channel=0, 1, 2 with 0: red, 1: green, 2: blue)
```

Berechnet das Histogramm des Layers **layer** des geladenen Bildes und erzeugt die Datei histo_[Kanalname].dat im Arbeitsverzeichnis.
layer = 0, 1 oder 2 mit 0=rot, 1=grün, 2=blau

iadd

```
iadd filename
```

Addiert das Bild **filename** zum geladenen Bild.
Das Ergebnis wird ein Bild mit 32 Bit pro Kanal sein, wenn dies in den Einstellungen erlaubt ist

icc_assign

```
icc_assign profile
```

Weist dem aktuell geladenen Bild das im Argument angebenene ICC-Profil zu.
Eines der folgenden speziellen Argumente kann angegeben werden, um die jeweiligen eingebauten Profile zu verwenden: **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **working**, um das Arbeits-Mono- oder RGB-Farbprofil festzulegen (nur für Mono-Bilder), **linear**, oder es kann der Pfad zu einer ICC-Profildatei angegeben werden. Wenn ein eingebautes Profil angegeben wird und ein Schwarzweißbild geladen ist, wird das Grauprofil mit der entsprechenden TRC verwendet

icc_convert_to

```
icc_convert_to profile [intent]
```

Konvertiert das aktuell geladenen Bild zum angegebenen ICC-Profil.
Eines der folgenden speziellen Argumente kann angegeben werden, um die jeweiligen eingebauten Profile zu verwenden: **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **graysrgb**, **grayrec2020**, **graylinear* oder **working**, um das Arbeits-Mono- oder RGB-Farbprofil festzulegen (nur für Mono-Bilder), **linear**, oder es kann der Pfad zu einer ICC-Profildatei angegeben werden. Wenn ein eingebautes Profil angegeben wird und ein Schwarzweißbild geladen ist, wird das Grauprofil mit der entsprechenden TRC verwendet.

Ein zweites Argument kann angegeben werden um die Absicht der Farbtransformation anzugeben. Die sollte eine der folgenden sein: **perceptual** (Wahrnehmung), **relative** (für relativ farbmetrisch), **saturation** (Sättigung) or **absolute** (für absolut farbmetrisch)

icc_remove

```
icc_remove
```

Entfernt das ICC-Profil aus dem aktuellen Bild, sofern vorhanden

idiv

```
idiv filename
```

Teilt das geladene Bild im Speicher durch das im Argument **Dateiname** angegebene Bild.
Das Ergebnis wird ein Bild mit 32 Bit pro Kanal sein, wenn dies in den Einstellungen erlaubt ist.

Siehe auch FDIV

Verweis: *fdiv*

imul

```
imul filename
```

Multipliziert das geladene Bild im Speicher mit dem im Argument **Dateiname** angegebenen Bild.
Das Ergebnis wird ein Bild mit 32 Bit pro Kanal sein, wenn dies in den Einstellungen erlaubt ist

inspector

```
inspector
```

Teilt das aktuelle Bild in ein Neun-Panel-Mosaik, das die Bildecken und die Mitte für eine genauere Betrachtung zeigt (nur GUI/Benutzeroberfläche)

invght

```
invght -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -
↪sat] [channels]
```

Kehrt eine generalisierte hyperbolische Streckung um. Es bietet die umgekehrte Transformation von GHT. Wenn es mit denselben Parametern versehen wird, macht es einen GHT-Befehl rückgängig und kehrt möglicherweise zu einem linearen Bild zurück. Es kann für Negativbilder auch auf die identische Weise wie GHT funktionieren

Verweise: *ght*

invmodasinh

```
invmodasinh -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -sat]␣
↪[channels]
```

Kehrt eine modifizierte Arcsinh-Streckung um. Es bietet die umgekehrte Transformation von MODASINH. Wenn es mit denselben Parametern versehen wird, macht es einen MODASINH-Befehl rückgängig und kehrt möglicherweise zu einem linearen Bild zurück. Es kann auch bei Negativbildern auf die gleiche Weise wie MODASINH funktionieren

Verweise: *modasinh*

invmtf

```
invmtf low mid high [channels]
```

Kehrt eine Mitteltonübertragungsfunktion um. Es bietet die umgekehrte Transformation von MTF. Wenn es mit denselben Parametern versehen wird, macht es einen MTF-Befehl rückgängig und kehrt möglicherweise zu einem linearen Bild zurück. Es kann bei Negativbildern auch auf die gleiche Weise wie MTF funktionieren

Verweis: *mtf*

isub

```
isub filename
```

Subtrahiert das geladene Bild im Speicher von dem im Argument **filename** angegebenen Bild.
Das Ergebnis wird ein Bild mit 32 Bit pro Kanal sein, wenn dies in den Einstellungen erlaubt ist, es können also negative Werte gespeichert werden. Um negative Werte zu beschneiden, verwenden Sie 16 Bit Ausgangsbilder oder beschneiden Sie die Bilder mit dem THRESHLO-Befehl

Verweis: *threshlo*

jsonmetadata

```
jsonmetadata FITS_file [-stats_from_loaded] [-nostats] [-out=]
```

Gibt Metadaten und Statistiken des aktuell geladenen Bildes in JSON-Form aus. Der Dateiname ist erforderlich, auch wenn das Bild bereits geladen ist. Bilddaten dürfen nicht aus der Datei gelesen werden, wenn es sich um das aktuell geladene Bild handelt und die Option **-stats_from_loaded** übergeben wird. Die Statistik kann mit der Option **-nostats** deaktiviert werden. Eine Datei mit den JSON-Daten wird mit dem Standard-Dateinamen "$(FITS_file_without_ext).json" erstellt und kann mit der Option **-out=** geändert werden

Lichtkurve

```
light_curve sequencename channel [-autoring] { -at=x,y | -wcs=ra,dec } { -refat=x,y | -
↪refwcs=ra,dec } ...
light_curve sequencename channel [-autoring] -ninastars=file
```

Analysiert mehrere Sterne mit der Aperturphotometrie in einer Bildsequenz und erstellt eine Lichtkurve für einen Stern, die durch die anderen kalibriert wird. Die ersten Koordinaten, in Pixeln, wenn **-at=** verwendet wird, oder in Grad, wenn **-wcs=** verwendet wird, sind für den Stern, dessen Licht aufgezeichnet werden soll, die anderen für die Vergleichssterne.

Alternativ kann eine Liste von Ziel- und Referenzsternen im Format der Sternliste des NINA Exoplaneten-Plugins mit der Option **-ninastars=** übergeben werden. Siril prüft, ob alle Referenzsterne verwendet werden können, bevor es sie tatsächlich verwendet. Es wird eine Datendatei im aktuellen Verzeichnis mit dem Namen light_curve.dat erstellt, Siril zeichnet das Ergebnis in ein PNG-Bild, falls verfügbar

Die Ringradien für den Ring können entweder in den Einstellungen konfiguriert werden oder auf einen Faktor der FWHM des Referenzbildes gesetzt werden, wenn **-autoring** übergeben wird. Diese Autoring-Größen sind das 4,2-fache und das 6,3-fache der FWHM für den inneren bzw. äußeren Radius.

Siehe auch den Befehl **setphot**, um auf die gleiche Weise die Größe des Blendenradius einzustellen.

Siehe auch SEQPSF für Operationen an einem einzelnen Stern

Verweis: *seqpsf*

limit

```
limit { -clip | -posrescale | -rescale }
```

Begrenzt Pixelwerte in 32-Bit-Bildern auf den Bereich von 0,0 bis 1,0. Dieser Befehl gilt nicht für 16-Bit-Bilder, da es keine Werte außerhalb des Bereichs geben kann. Die Bereichsbegrenzung kann auf eine der folgenden Arten erfolgen:

-clip: Diese Option setzt einfach alle negativen Pixel auf 0,0 und alle Pixel mit einem Wert > 1,0 auf 1,0.

-posrescale: Diese Option skaliert alle positiven Pixelwerte so, dass der Maximalwert 1,0 beträgt, und schneidet alle negativen Pixel auf 0,0 ab. Bei 3-Kanal-Bildern wird auf alle Kanäle der gleiche Skalierungsfaktor angewendet. Wenn der maximale Pixelwert bereits <= 1,0 ist, werden negative Pixel trotzdem abgeschnitten, aber auf positive Pixel wird kein Skalierungsfaktor angewendet.

-rescale: Wenn diese Option verwendet wird, wird bei negativen Pixelwerten ein konstanter Wert zu allen Pixelwerten des Bildes addiert, sodass der Mindestwert 0,0 beträgt. Wenn der maximale Pixelwert dann > 1,0 ist, wird ein Skalierungsfaktor angewendet, sodass der maximale Pixelwert auf 1,0 skaliert wird.

Beachten Sie, dass die Optionen **-rescale** und **-porescale** bei einem oder mehreren extremen Ausreißern (zum Beispiel aufgrund fehlerhafter Pixel) zu unerwarteten Ergebnissen führen können. Dies kann gemildert werden, indem zunächst eine kosmetische Korrektur auf das Bild angewendet wird.

linear_match

```
linear_match reference low high
```

Berechnet eine lineare Funktion zwischen einem **reference**-Bild und dem geladenen Bild im Speicher und wendet sie an.

Der Algorithmus ignoriert alle Referenzpixel, deren Werte außerhalb des Bereichs [**low**, **high**] liegen

link

```
link basename [-date] [-start=index] [-out=]
```

Identisch mit CONVERT, konvertiert jedoch nur FITS-Dateien, die im aktuellen Arbeitsverzeichnis gefunden werden. Dies ist nützlich, um Konvertierungen von JPEG-Ergebnissen oder anderen Dateien zu vermeiden, die möglicherweise im Verzeichnis landen. Das zusätzliche Argument **-date** ermöglicht die alphanumerische Sortierung von Dateien nach ihrem DATE-OBS-Wert statt nach ihrem Namen

Verweise: *convert*

linstretch

```
linstretch -BP= [-sat] [-clipmode=] [channels] [-clipmode=]
```

Streckt das Bild linear auf einen neuen Schwarzpunkt BP.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: dies können R, G, B, RG, RB oder GB sein. Die Standardeinstellung ist alle Kanäle

Optional kann der Parameter **-sat** verwendet werden, um die lineare Streckung auf den Bildsättigungskanal anzuwenden. Dieses Argument funktioniert nur, wenn alle Kanäle ausgewählt sind. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend

livestack

```
livestack filename
```

Verarbeitet das bereitgestellte Bild für das Live-Stacking. Nur nach START_LS möglich. Der Prozess umfasst das Kalibrieren der eingehenden Datei, sofern in START_LS konfiguriert, das Debayern, wenn es sich um ein OSC-Image handelt, das Registrieren und Stacken. Das temporäre Ergebnis wird in der Datei live_stack_00001.fit gespeichert, bis eine neue Option zum Ändern hinzugefügt wird

Verweis: *start_ls*

Warnung

Beachten Sie, dass die Live-Stacking-Befehle Siril in einen Zustand versetzen, in dem es nicht in der Lage ist, andere Befehle zu verarbeiten. Nach START_LS können nur LIVESTACK, STOP_LS und EXIT aufgerufen werden, bis STOP_LS aufgerufen wird, um Siril wieder in den normalen, nicht-Live-Stacking-Zustand zu versetzen.

laden

```
load filename[.ext]
```

Lädt das Bild **filename** aus dem aktuellen Arbeitsverzeichnis, welches dann das aktuell geladene Bild wird welches in vielen Einzelbildbefehlen verwendet wird.
Es wird zuerst versucht **filename**, dann **filename**.fit, abschließend **filename**.fits und anschließend alle unterstützten Bildformate zu laden.
Dieses Schema ist auf jeden Siril-Befehl anwendbar, der das Lesen von Dateien beinhaltet

load_seq

```
load_seq sequencename[.ext]
```

Loads the sequence **sequencename** from the current working directory, which becomes the 'currently loaded sequence'. While Siril sequence commands require the sequence name to be specified, it is necessary to load a sequence in order to read its metadata using Python scripting

log

```
log
```

Berechnet eine logarithmische Skalierung und wendet sie auf das geladene Bild an, indem die folgende Formel verwendet wird: log(1 – (Wert – Min) / (Max – Min)), wobei Min und Max der minimale und maximale Pixelwert für den Kanal sind

ls

```
ls
```

Listet Dateien und Verzeichnisse im Arbeitsverzeichnis auf

Dieser Befehl ist nur auf Unix-ähnlichen Systemen verfügbar. Für den entsprechenden Befehl unter Microsoft Windows siehe *dir*.

makepsf

```
makepsf clear
makepsf load filename
makepsf save [filename]
makepsf blind [-l0] [-si] [-multiscale] [-lambda=] [-comp=] [-ks=] [-savepsf=]
makepsf stars [-sym] [-ks=] [-savepsf=]
makepsf manual { -gaussian | -moffat | -disc | -airy } [-fwhm=] [-angle=] [-ratio=] [-
↪beta=] [-dia=] [-fl=] [-wl=] [-pixelsize=] [-obstruct=] [-ks=] [-savepsf=]
```

Erzeugt eine PSF zur Verwendung mit der Dekonvolution, eine der drei Methoden die durch die Befehle RL, SB oder WIENER erstellt werden. Als erstes Argument muss eine der folgenden Optionen angegeben werden: **clear** (löscht die vorhandene PSF), **load** (lädt eine PSF aus einer Datei), **save** (speichert die aktuelle PSF), **blind** (blinde Schätzung der PSF), **stars** (erzeugt eine PSF auf der Grundlage von gemessenen Sternen aus dem Bild) oder **manual** (erzeugt eine PSF manuell auf der Grundlage einer Funktion und von Parametern).

Bei Verwendung des Arguments **clear** sind keine zusätzlichen Argumente erforderlich.

Um ein zuvor gespeichertes PSF zu laden, benötigt das Argument **load** den PSF-*Dateinamen* als zweites Argument. Dieser kann in jedem beliebigen Format vorliegen, für das Siril kompiliert wurde, muss aber quadratisch sein und sollte idealerweise eine ungerade Seitenlänge haben.

Zum Speichern des aktuellen PSF wird das Argument **save** verwendet. Optional kann ein Dateiname angegeben werden (dieser muss eine der Erweiterungen ".fit", ".fits", ".fts" oder ".tif" haben), wird jedoch keiner angegeben, so wird das PSF anhand des Namens der geöffneten Datei oder Sequenz benannt.

Für **blind** können die folgenden optionalen Argumente angegeben werden: **-l0** verwendet die l0-Abstiegsmethode, **-si** verwendet die Methode der spektralen Unregelmäßigkeit, **-multiscale** konfiguriert die l0-Methode für eine PSF-Schätzung mit mehreren Skalierungen, **-lambda=** gibt die Regularisierungskonstante an.

Für PSF von gefundenen Sternen ist der einzige optionale Parameter **-sym**, mit dem die PSF als symmetrisch konfiguriert wird.

Für eine manuelle PSF muss eine der Optionen **-gaussian**, **-moffat**, **-disc** oder **-airy** angegeben werden, um die PSF-Funktion zu spezifizieren. Für Gauß- oder Moffat-PSFs können die optionalen Argumente **-fwhm=**, **-angle=** und **-ratio=** angegeben werden. Für Moffat-PSFs kann auch das optionale Argument **-beta=** angegeben werden. Wenn diese Werte nicht angegeben werden, werden die gleichen Werte wie im Dekonvolutionsdialog verwendet. Für Scheiben-PSFs ist nur das Argument **-fwhm=** erforderlich, das für diese Funktion verwendet wird, um den *Durchmesser* der PSF festzulegen. Für Airy-PSFs (Beugungsscheibchen) können die folgenden Argumente angegeben werden: **-dia=** (legt den Durchmesser der Teleskopöffnung fest), **-fl=** (legt die Teleskopbrennweite fest), **-wl=** (legt die Wellenlänge fest, für die das Airy-Beugungsmuster berechnet werden soll), **-pixelsize=** (legt die Sensorpixelgröße fest), **-obstruct=** (legt die zentrale Obstruktion als Prozentsatz der Gesamtöffnung fest). Wenn diese Parameter nicht angegeben werden, ist die Wellenlänge standardmäßig 525 nm und die zentrale Obstruktion standardmäßig 0 %. Siril versucht, die anderen Parameter aus dem geöffneten Bild auszulesen, aber manche Bildbearbeitungsprogramme stellen möglicherweise nicht alle Parameter zur Verfügung, so dass Sie schlechte Ergebnisse erhalten. Sie werden aus Erfahrung lernen, welche Metadaten Sie bei Ihrer speziellen Bildbearbeitungssoftware getrost weglassen können.

Für jede der oben genannten Optionen zur PSF-Erzeugung kann das optionale Argument **-ks=** angegeben werden, um die PSF-Dimension festzulegen, und das optionale Argument **-savepsf=filename** kann verwendet werden, um das erzeugte PSF zu speichern: ein Dateiname muss angegeben werden, und es gelten dieselben Anforderungen an die Dateinamenerweiterung wie für **makepsf save filename**

Verweise: *psf*, *rl*, *sb*, *wiener*

merge

```
merge sequence1 sequence2 [sequence3 ...] output_sequence
```

Fügt mehrere Sequenzen desselben Typs (FITS-Bilder, FITS-Sequenz oder SER) und derselben Bildeigenschaften zu einer neuen Sequenz mit dem Basisnamen **newseq** zusammen, die im aktuellen Arbeitsverzeichnis mit demselben Typ erstellt wird. Die Eingabesequenzen können sich in verschiedenen Verzeichnissen befinden und entweder als absoluter oder relativer Pfad, mit dem genauen .seq-Namen oder nur mit dem Basisnamen mit oder ohne abschließendem '_' angegeben werden.

merge_cfa

```
merge_cfa file_CFA0 file_CFA1 file_CFA2 file_CFA3 bayerpattern
```

Erzeugt ein Bayer-Farbbild aus 4 separaten Bildern, die die Daten der Bayer-Unterkanäle CFA0, CFA1, CFA2 und CFA3 enthalten. (Der entsprechende Befehl zur Aufteilung des CFA-Musters in Unterkanäle lautet **split_cfa**). Diese Funktion kann als Teil eines Arbeitsablaufs verwendet werden, bei dem die einzelnen Bayer-Farbkanäle vor dem Debayern verarbeitet werden. Der fünfte Parameter **bayerpattern** gibt das Bayer-Matrixmuster an, das neu erstellt werden soll: *bayerpattern* sollte einer der Werte 'RGGB', 'BGGR', 'GRBG' oder 'GBRG' sein

mirrorx

```
mirrorx [-bottomup]
```

Spiegelt das Bild um die horizontale Achse. Die Option **-bottomup** spiegelt das Bild nur, wenn es nicht bereits "von-unten-nach-oben" ist

mirrorx_single

```
mirrorx_single image
```

Spiegelt das Bild bei Bedarf um die horizontale Achse (wenn es nicht bereits von unten nach oben ausgerichtet ist). Verwendet den Bilddateinamen als Argument, wodurch das Lesen von Bilddaten vollständig vermieden werden kann, wenn keine Spiegelung erforderlich ist. Das Bild wird überschrieben, falls eine Spiegelung durchgeführt wurde

mirrory

```
mirrory
```

Spiegelt das Bild um die vertikale Achse

modasinh

```
modasinh -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -independent | -sat]␣
↪[channels]
```

Modifizierte ArcSinH-Streckung auf der Grundlage der Arbeit des Teams von ghsastro.co.uk.

Das Argument **-D=** bestimmt die Stärke der Streckung, die zwischen 0 und 10 liegt. Dies ist das einzige obligatorische Argument. Mit den folgenden optionalen Argumenten kann die Streckung weiter angepasst werden:

LP definiert einen schattenerhaltenden Bereich zwischen 0 und SP, in dem die Streckung linear erfolgt und die Schattendetails erhalten bleiben;

SP definiert den Symmetriepunkt der Streckung, der zwischen 0 und 1 liegt und an dem die Streckung am stärksten ist;

HP definiert einen Bereich zwischen HP und 1, in dem die Streckung linear erfolgt, wodurch die Details der Glanzlichter erhalten bleiben und eine Aufblähung der Sterne verhindert wird.

Ohne diese Angabe haben LP und SP den Standardwert 0,0 und HP den Standardwert 1,0.

Ein optionales Argument (entweder **-human**, **-even** oder **-independent**), kann übergeben werden, um entweder menschlich-gewichtete oder gleichmäßig gewichtete Luminanz oder unabhängige Farbkanäle für Farbstreckungen auszuwählen. Das Argument wird bei Monobildern ignoriert. lternativ gibt das Argument **-sat** an, dass die Streckung anhand der Bildsättigung erfolgt - das Bild muss farbig sein und alle Kanäle müssen ausgewählt sein, damit dies funktioniert.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: Dies können R, G, B, RG, RB oder GB sein. Der Standardwert sind alle Kanäle. Der Clip-Modus kann mit dem Argument **-clipmode=** eingestellt werden: Werte wie **clip**, **rescale**, **rgbblend** oder **globalrescale** werden akzeptiert und der Standardwert ist rgbblend.

mtf

```
mtf low mid high [channels]
```

Wendet die Mittelton-Übertragungsfunktion auf das aktuell geladene Bild an.

Es werden drei Parameter benötigt: **low**, **midtones** und **high**, wobei der Parameter für die Mitteltonbalance eine nichtlineare Histogrammstreckung im Bereich [0,1] definiert. Für eine automatische Ermittlung der Parameter siehe AUTOSTRETCH.

Optional kann der Parameter **[channels]** verwendet werden, um die Kanäle anzugeben, auf die die Streckung angewendet werden soll: dies können R, G, B, RG, RB oder GB sein. Die Standardeinstellung ist alle Kanäle

Verweise: *autostretch*

neg

```
neg
```

Ändert die Pixelwerte des aktuell geladenen Bildes in eine negative Ansicht. Beispielsweise "1-value" für 32-Bit, "65535-value" für 16-Bit. Dies ändert nicht den Anzeigemodus

new

```
new width height nb_channel [filename]
```

Erstellt eine neue, schwarze Bilddatei der Größe **width** x **height**.
Optional kann der Parameter **[filename]** verwendet werden, um den Namen der neuen Datei zu spezifizieren.

Das Bild ist im 32-Bit-Format und enthält **nb_channel** Kanäle, wobei **nb_channel** 1 oder 3 ist. Es wird nicht gespeichert, sondern wird zum geladenen Bild, das angezeigt wird und anschließend gespeichert werden kann

nozero

```
nozero level
```

Ersetzt Null-Werte durch **level**-Werte. Das ist sinnvoll vor einer idiv- oder fdiv-Operation, besonders für 16-Bit-Bilder

offline

```
offline
```

Stellt Siril auf Offline-Modus. In diesem Modus sind Netzwerkfunktionen wie Remote-Katalogabfragen, Aktualisierung von Git-Repositories usw. nicht verfügbar. Zwischengespeicherte Daten sind jedoch weiterhin zugänglich

offset

```
offset value
```

Addiert den konstanten Wert **value** (angegeben in ADU) zum aktuellen Bild. Diese Konstante kann auch einen negativen Wert haben.

Im 16-Bit-Modus werden die Werte von Pixeln, die außerhalb von [0, 65535] liegen, abgeschnitten. Im 32-Bit-Modus findet keine Beschneidung statt

Online

```
online
```

Stellt Siril auf Online-Modus. In diesem Modus sind Netzwerkfunktionen wie Remote-Katalogabfragen, Aktualisierung von Git-Repositories usw. erlaubt

parse

```
parse str [-r]
```

Parsed the String **str** basierend auf den Informationen im Header des aktuelle geladenen Bildes. Hauptzweck dieses Befehls ist es, das Path-Parsing von Header-Schlüsseln zu debuggen welches in anderen Befehlen genutzt werden kann.

Option **-r** specifies the string is to be interpreted in read mode. In read mode, all wildcards defined in string **str** are used to find a file name matching the pattern. Otherwise, default mode is write mode and wildcards, if any, are removed from the string to be parsed.

Wenn **str** mit dem Präfix *$def* beginnt, wird es als reserviertes Schlüsselwort erkannt und in den in gui_prepro.dark_lib, gui_prepro.flat_lib, gui_prepro.bias_lib oder gui_prepro.stack_default gespeicherten Strings für *$defdark*, *$defflat*, *$defbias* bzw. *$defstack* gesucht.

Das Schlüsselwort *$seqname$* kann auch verwendet werden, wenn eine Sequenz geladen ist

pcc

```
pcc [-limitmag=[+-]] [-catalog=] [-bgtol=lower,upper]
```

Führt die photometrische Farbkalibrierung auf dem geladenen, astrometrisch gelösten Bild durch.

Die Grenzgröße der Sterne wird automatisch aus der Größe des Bildfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.

Der verwendete Sternkatalog ist standardmäßig NOMAD, er kann durch Angabe von **-catalog=apass**, **-catalog=localgaia** oder **-catalog=gaia** geändert werden. Falls NOMAD lokal installiert ist, kann der Remote-NOMAD (vollständige Version) durch Angabe von **-catalog=nomad** erzwungen werden

Die Toleranz für Außreißer der Hintergrundreferenz kann in Sigma-Einheiten mit **-bgtol=lower,upper** angegeben werden: die Standardwerte sind -2,8 und +2,0

platesolve

```
platesolve [-force] [image_center_coords] [-focal=] [-pixelsize=]
platesolve ... [-noflip] [-downscale] [-order=] [-radius=] [-disto=]
platesolve ... [-limitmag=[+-]] [-catalog=] [-nocrop]
platesolve ... [-localasnet [-blindpos] [-blindres]]
```

Berechnet eine astrometrische Lösung für das geladene Bild.

Wenn das Bild bereits eine astrometrische Lösung besitzt, wird nichts getan, es sei denn, das Argument **-force** wird übergeben, um eine neue Lösung zu erzwingen. Wenn WCS oder andere Bild-Metadaten fehlerhaft sind oder fehlen, müssen Argumente übergeben werden:

Die ungefähren Koordinaten des Bildmittelpunkts können in Dezimalgraden oder Grad/Stunde-Minute-Sekunde-Werten (J2000 mit Doppelpunkten) angegeben werden, wobei die Werte für Rektaszension und Deklination durch ein Komma oder ein Leerzeichen getrennt sind (für astrometry.net nicht obligatorisch).

Brennweite und Pixelgröße können mit **-focal=** (in mm) und **-pixelsize=** in (in µm) übergeben werden, wobei die Werte aus dem Bild bzw. den Einstellungen überschrieben werden. Siehe auch unter Optionen zum blinden Lösen mit lokalem Astrometry.net

Wenn **-noflip** nicht angegeben ist, wird das Bild gespiegelt, wenn es als auf dem Kopf stehend erkannt wird.

Für eine schnellere Erkennung von Sternen in großen Bildern ist eine Verkleinerung des Bildes mit **-downscale** möglich.

Bei der Lösung können Verzerrungen nach der SIP-Konvention mit Polynomen bis zur Ordnung 5 berücksichtigt werden. Standardmäßig wird die kubische Verzerrung aus den Astrometrie-Einstellungen genutzt. Dies kann mit der Option **-order=** geändert werden, die einen Wert zwischen 1 und 5 angibt.

Wenn Sie lokale Siril-Solver-Kataloge oder lokales Astrometry.net verwenden und die anfängliche Lösung nicht erfolgreich ist, sucht der Solver nach einer Lösung innerhalb eines Radiuskegels, der mit der Option **-radius=** angegeben wird. Wenn kein Wert übergeben wird, wird der Suchradius aus den Astrometrieeinstellungen übernommen. Die Siril-Nähesuche kann durch Übergeben eines Werts von 0 deaktiviert werden. (kann für Astrometry.net nicht deaktiviert werden).

Mit der Option **-disto=** können Sie die aktuelle astrometrische Lösung als Verzerrungsdatei speichern.

Bilder können entweder mit Siril unter Verwendung eines Sternkatalogs und des globalen Registrierungsalgorithmus oder mit dem lokalen solve-field-Befehl von astrometry.net (aktiviert mit **-localasnet**) astrometrisch gelöst werden.

Optionen des Siril Lösers:

Die Grenzgröße der Sterne, die für die Plattenlösung verwendet werden, wird automatisch aus der Größe des Sichtfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.

Die Wahl des Sternkatalogs erfolgt automatisch, es sei denn, die Option **-catalog=** wird übergeben: Wenn lokale

Kataloge installiert sind, werden sie verwendet, andernfalls erfolgt die Wahl auf der Grundlage des Sichtfelds und der Grenzgröße. Wenn die Option übergeben wird, erzwingt sie die Verwendung des im Argument angegebenen Katalogs, mit den möglichen Werten: tycho2, nomad, localgaia, gaia, ppmxl, brightstars, apass.

Wenn das berechnete Bildfeld größer als 5 Grad ist, wird die Sternerkennung auf eine beschnittenen Bereich um das Bildzentrum begrenzt, falls nicht die Option **-nocrop** verwendet wird.

Optionen des Astrometry.net Lösers:

Die Übergabe der Optionen **-blindpos** und/oder **-blindres** ermöglicht die blinde Lösung nach Position bzw. nach Sampling. Sie können diese verwenden, wenn Sie ein Bild mit einem völlig unbekannten Bildausschnitt und einem unbekannten Sampling lösen

pm

```
pm "expression" [-rescale [low] [high]] [-nosum]
```

Dieser Befehl wertet den im Argument angegebenen Ausdruck wie im PixelMath-Tool aus. Der vollständige Ausdruck muss in Anführungszeichen stehen, und die Variablen (d. h. die Bildnamen ohne Erweiterung, die sich in diesem Fall im Arbeitsverzeichnis befinden) müssen von dem Token $ umgeben sein, z. B. "$Bild1$ * 0,5 + $Bild2$ * 0,5". Es können maximal 10 Bilder in dem Ausdruck verwendet werden.

Das Bild kann mit der Option **-rescale**, gefolgt von den Werten **low** und **high** im Bereich [0, 1], neu skaliert werden. Wenn keine niedrigen (low) und hohen (hight) Werte angegeben werden, werden die Standardwerte auf 0 und 1 gesetzt. Ein weiteres optionales Argument, **-nosum**, weist Siril an, Belichtungszeiten nicht zu summieren. Dies wirkt sich auf FITS-Schlüsselwörter wie LIVETIME und STACKCNT aus

Profil

```
profile -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] { [-savedat] | [-filename=] } [-
↪layer=] [-width=] [-spacing=] ["-title=My Plot"]
```

Erzeugt ein Intensitätsprofil zwischen 2 Punkten im Bild, auch bekannt als *Schnitt*. Die Argumente können in beliebiger Reihenfolge angegeben werden. Die Argumente **-to=x,y** und **-from=x,y** sind obligatorisch.

Das Argument **-layer={red | green | blue | lum | col}** gibt an, welcher Kanal (oder Luminanz oder Farbe) gezeichnet werden soll, wenn das Bild farbig ist. Es kann auch in Verbindung mit der Option **-tri** verwendet werden, die drei parallele, gleichabständige Profile erzeugt, die jeweils durch **-spacing=** Pixel voneinander getrennt sind. Beachten Sie jedoch, dass bei Tri-Profilen die Option **col** genauso behandelt wird wie **lum**.

Mit der Option **-cfa** wird der CFA-Modus ausgewählt, der 4 Profile erzeugt: 1 für jeden CFA-Kanal in einem Bild mit Bayer-Muster. Diese Option kann nicht mit Farbbildern oder Schwarzweißbildern ohne Bayer-Muster verwendet werden und kann nicht gleichzeitig mit der Option **-tri** verwendet werden.

Die Option **-arcsec** bewirkt, dass die x-Achse die Entfernung in arcsec anzeigt, wenn die erforderlichen Metadaten verfügbar sind. Wenn die Option nicht angegeben ist oder keine Metadaten vorhanden sind, wird die Entfernung in Pixel-Einheiten angezeigt.

Das Argument **-savedat** bewirkt, dass die Datendateien gespeichert werden: Der Dateiname wird in das Protokoll geschrieben. Alternativ kann das Argument **-filename=** verwendet werden, um einen Dateinamen anzugeben, in den die Datendatei geschrieben werden soll. (Die Option **-filename=** impliziert **-savedat**.)

Das Argument **"-title=Mein Titel "** setzt einen benutzerdefinierten Titel "Mein Titel"

psf

```
psf [channel]
```

Führt eine PSF (Point Spread Function) für den ausgewählten Stern durch und zeigt die Ergebnisse an. Im Headless-Betrieb kann die Auswahl mit BOXSELECT in Pixeln angegeben werden. Falls angegeben, wählt das Argument **channel** den Bildkanal aus, auf dem der Stern analysiert werden soll. Es kann bei monochromen Bildern oder bei der Ausführung über die grafische Benutzeroberfläche weggelassen werden, wenn einer der Kanäle in der Ansicht aktiv ist

Links: *boxselect*

pwd

```
pwd
```

Zeigt das aktuelle Arbeitsverzeichnis an

pyscript

```
pyscript [-async] scriptname.py [script_argv]
```

Executes a Siril python script

An optional argument **-async** may be provided. If it is provided it **must** be the first argument. This allows the script launched by pyscript to run in the background without blocking a script that calls it. This is expected to be useful for launcher-type scripts. The default is for the script to block until completion if pyscript is run as part of a script, but not to block if run from the GUI command entry (so the -async argument makes no difference when running like that).

The script name must be provided next (or first, if **-async** is not provided). If it is not found in the current working directory, the user-defined script paths specified in Preferences and the local siril-scripts repository will be searched. All subsequent arguments will be treated as script arguments and passed to the script as its argument vector. Note that the specific script must incorporate support for reading input from the argument vector

register

```
register sequencename [-2pass] [-selected] [-prefix=] [-scale=]
register sequencename ... [-layer=] [-transf=] [-minpairs=] [-maxstars=] [-nostarlist] [-
↪disto=]
register sequencename ... [-interp=] [-noclamp]
register sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
```

Sucht und führt optional geometrische Transformationen an Bildern der im Argument angegebenen Sequenz durch, sodass sie auf das Referenzbild gelegt werden können. Dieser Algorithmus verwendet Sterne zur Registrierung und funktioniert nur mit Deep-Sky-Bildern. Optionen zur Sternerkennung können mit **SETFINDSTAR** oder dem Dialog *Dynamische PSF* geändert werden.

Alle Bilder der Sequenz werden registriert, es sei denn, die Option **-selected** wird übergeben, in diesem Fall werden die ausgeschlossenen Bilder nicht verarbeitet

Die Option **-2pass** berechnet nur die Transformationen, generiert aber keine transformierten Bilder. **-2pass** fügt dem Algorithmus einen Vorab-Durchlauf hinzu, um vor der Berechnung der Transformationen basierend auf Bildqualität und Rahmung ein gutes Referenzbild zu finden. Um nach diesem Durchlauf transformierte Bilder zu generieren, verwenden Sie SEQAPPLYREG.

Wenn erstellt, beginnt der Name der Ausgabesequenz mit dem Präfix „r_“, sofern mit der Option **-prefix=** nichts anderes angegeben ist. Die Ausgabebilder können neu skaliert werden, indem ein **-scale=**-Argument mit einem Gleitkommawert zwischen 0,1 und 3 übergeben wird.

Optionen zu Bildtransformationen:

Die Erkennung erfolgt auf der grünen Ebene für Farbbilder, sofern nicht durch die Option **-layer=** mit einem Argument im Bereich von 0 bis 2 für Rot bis Blau etwas anderes angegeben ist.

-transf= gibt die Verwendung von **shift**-, **similarity**-, **affine**- oder **homography**-Transformationen (Standard) an.

Die Option **-minpairs=** gibt die Mindestanzahl von Sternpaaren an, die ein Bild mit dem Referenzbild haben muss, andernfalls wird das Bild ausgelassen und aus der Sequenz ausgeschlossen.

Die Option **-maxstars=** gibt die maximale Anzahl der Sterne an, die in jedem Bild gefunden werden sollen (muss zwischen 100 und 2000 liegen). Mit mehr Sternen kann eine genauere Registrierung berechnet werden, die jedoch mehr Zeit in Anspruch nimmt.

-nostarlist verhindert das Speichern der Sternliste.

-disto= verwendet Verzerrungsterme aus einer vorherigen astrometrischen Lösung (mit einer SIP-Ordnung> 1). Als Parameter wird entweder **image** verwendet, um die im aktuell geladenen Bild enthaltene Lösung zu verwenden, **file**

gefolgt vom Pfad zum Bild, das die Lösung enthält, oder **master**, um automatisch den passenden Verzerrungsmaster zu jedem Bild zu laden. Bei Verwendung dieser Option werden die Polynome sowohl zum Korrigieren der Sternpositionen vor der Berechnung der Transformation als auch zum Entzerren der Bilder beim Exportieren der Ausgabebilder verwendet.

Optionen zur Bildtransformation:

Standardmäßig werden Transformationen angewendet, um die Bilder mittels Interpolation zu registrieren.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden.

Optionen zum Drizzeln:

Andernfalls können die Bilder mit dem HST-Drizzle-Algorithmus exportiert werden, indem das Argument **-drizzle** übergeben wird, das die folgenden zusätzlichen Optionen annehmen kann:

-pixfrac= legt den Pixelanteil fest (Standard = 1,0).

Das Argument **-kernel=** legt den Drizzle-Kernel fest und muss von einem der folgenden Argumente gefolgt werden: **point**, **turbo**, **square**, **gaussian**, **lanczos2** oder **lanczos3**. Der Standardwert ist **square**.

Das Argument **-flat=** gibt ein Master-Flat an, um die gedrizzelten Eingabepixel zu gewichten (Standard ist kein Flat).

Hinweis: Wenn Sie **-drizzle** auf Bilder anwenden, die mit einer Farbkamera aufgenommen wurden, dürfen die Eingabebilder nicht debayert werden. In diesem Fall erfolgt die Sternerkennung immer auf den grünen Pixeln

Verweise: *setfindstar*, *psf*, *seqapplyreg*

Skripte neu laden

```
reloadscripts
```

Lädt den Skriptordner neu ein und aktualisiert das Scriptmenu. Beachten Sie, dass dieser Befehle im Headless-Modus (d.h. über siril-cli) eventuell nicht benutzt werden kann und beachten Sie außerdem, dass die Aktualisierung asynchron erfolgt (d.h. der Befehl initiiert die Aktualisierung, wartet aber nicht auf die Fertigstellung)

requires

```
requires min_version [obsolete_version]
```

Gibt einen Fehler zurück, wenn die Version von Siril älter ist als die im ersten Argument übergebene erforderliche

Mindestversion. Optional wird ein zweites Argument für die Siril-Version verwendet, bei der das Skript veraltet ist: Gibt einen Fehler zurück, wenn die Version von Siril **neuer oder gleich** der im zweiten Argument übergebenen Version ist.

Beispiel: *requires 1.2.0 1.4.0* ermöglicht die Ausführung des Skripts für alle Versionen der 1.2.x- und 1.3.x-Reihen, jedoch nicht für Versionen vor 1.2.0 oder für Version 1.4.0 oder spätere Versionen.

resample

```
resample { factor | -width= | -height= | -maxdim= } [-interp=] [-noclamp]
```

Führt eine Neuskalierung des Bildes durch, entweder mit einem Faktor **factor** oder für die Zielbreite oder -höhe, die durch **-width=**, **-height=** oder **-maxdim=** vorgegeben wird. Dies wird im Allgemeinen verwendet, um die Größe von Bildern zu ändern: Ein Faktor von 0,5 teilt die Größe durch 2. Das Argument **-maxdim** kann verwendet werden, um die längste Dimension des Bildes auf eine festgelegte Größe zu ändern, was nützlich sein kann, um Bilder für bestimmte Websites zu optimieren, z. B. Social-Media-Websites.

In der grafischen Benutzeroberfläche können wir sehen, dass mehrere Interpolationsalgorithmen vorgeschlagen werden.

Die Pixelinterpolationsmethode kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **ne**[arest] (nächster), **cu**[bic] (kubisch), **la**[nczos4], **li**[near], **ar**[ea] (Bereich)}.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden

rgbcomp

```
rgbcomp red green blue [-out=result_filename] [-nosum]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename] [-nosum]
```

Erstellt eine RGB-Komposition aus drei unabhängigen Bildern oder eine LRGB-Komposition aus dem optionalen Luminanzbild und drei monochromen Bildern oder einem Farbbild. Das Ergebnisbild heißt composed_rgb.fit oder composed_lrgb.fit, es sei denn, es wird ein anderer Name im optionalen Argument angegeben. Ein weiteres optionales Argument, **-nosum**, weist Siril an, Belichtungszeiten nicht zu summieren. Dies wirkt sich auf FITS-Schlüsselwörter wie LIVETIME und STACKCNT aus

rgradient

```
rgradient xc yc dR dalpha
```

Erzeugt zwei Bilder mit einer radialen Verschiebung (**dR** in Pixeln) und einer Rotationsverschiebung (**dalpha** in Grad) in Bezug auf den Punkt (**xc**, **yc**).

Zwischen diesen beiden Bildern haben die Verschiebungen die gleiche Amplitude, aber ein entgegengesetztes Vorzeichen. Die beiden Bilder werden dann addiert, um das endgültige Bild zu erhalten. Dieses Verfahren wird auch Larson-Sekanina-Filter genannt

rl

```
rl [-loadpsf=] [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Stellt ein Bild nach der Richardson-Lucy-Methode wieder her.

Optional kann eine PSF mit dem Argument **-loadpsf=filename** (erzeugt mit MAKEPSF) geladen werden.

Die Anzahl der Iterationen wird durch **-iters** festgelegt (der Standardwert ist 10).

Die Art der Regularisierung kann mit **-tv** für die Gesamtvariation oder **-fh** für die Frobenius-Norm der Hessian-Matrix festgelegt werden (die Vorgabe ist keine) und **-alpha=** gibt die Stärke der Regularisierung an (kleinerer Wert = stärkere Regularisierung, Vorgabe = 3000).

Standardmäßig wird die Methode des Gradientenabstiegs mit einer Standardschrittweite von 0,0005 verwendet, doch kann die multiplikative Methode mit **-mul** erzwungen werden.

Das Stopkriterium kann durch Angabe einer Anhaltegrenze mit **-stop=** aktiviert werden

Verweise: *psf*, *makepsf*

rmgreen

```
rmgreen [-nopreserve] [type] [amount]
```

Wendet einen Filter zur Reduzierung des chromatischen Rauschens an. Er entfernt den Grünstich im aktuellen Bild. Dieser Filter basiert auf dem SCNR-Filter von PixInsight und ist auch der gleiche Filter, der vom HLVG-Plugin in Photoshop verwendet wird.
Die Helligkeit bleibt standardmäßig erhalten, kann aber mit dem Schalter **-nopreserve** deaktiviert werden.

Typ kann die Werte 0 für durchschnittliche Neutralität, 1 für maximale Neutralität, 2 für maximale Maske, 3 für additive Maske annehmen, wobei der Standardwert 0 ist. Die letzten beiden können ein Argument **Betrag** annehmen, einen Wert zwischen 0 und 1, wobei der Standardwert 1 ist

Rotieren

```
rotate degree [-nocrop] [-interp=] [-noclamp]
```

Dreht das Bild um einen Winkel mit dem Wert **degree**. Die Option **-nocrop** kann hinzugefügt werden, um ein Beschneiden auf die Bildgröße zu vermeiden (es werden schwarze Ränder hinzugefügt).

Hinweis: Wenn eine Auswahl aktiv ist, d. h. durch Verwendung eines Befehls `boxselect` vor `rotate`, wird das resultierende Bild ein gedrehter Ausschnitt sein. In diesem besonderen Fall wird die Option **-nocrop** ignoriert, wenn sie übergeben wird.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet.
Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden

rotatePi

```
rotatePi
```

Dreht das geladene Bild um 180° um dessen Mitte. Dies ist gleichwertig mit dem Befehl "ROTATE 180" oder "ROTATE -180"

Verweis: *rotate*

satu

```
satu amount [background_factor [hue_range_index]]
```

Erhöht die Farbsättigung des geladenen Bildes. Versuchen Sie es iterativ, um beste Ergebnisse zu erzielen.

amount kann eine positive Zahl sein, um die Farbsättigung zu erhöhen, eine negative, um sie zu verringern, 0 würde nichts bewirken, 1 würde sie um 100% erhöhen

background_factor ist ein Faktor (Median + Sigma), der verwendet wird, um einen Schwellenwert festzulegen, bei dem nur die Pixel oberhalb dieses Wertes geändert werden. Dies ermöglicht es, dass das Hintergrundrauschen nicht farblich gesättigt wird, wenn es sorgfältig gewählt wird. Der Standardwert ist 1, mit 0 wird der Schwellenwert deaktiviert.

hue_range_index kann [0, 6] sein, was bedeutet: 0 für rosa bis orange, 1 für orange bis gelb, 2 für gelb bis cyan, 3 für cyan, 4 für cyan bis magenta, 5 für magenta bis rosa, 6 für alle (Standard)

save

```
save filename [-chksum]
```

Speichert das aktuelle Bild unter **filename**.fit (oder .fits, je nach Ihren Einstellungen, siehe SETEXT) im aktuellen Arbeitsverzeichnis. Das Bild bleibt geladen. **filename** kann einen Pfadnamen enthalten, solange das Verzeichnis bereits existiert. Die Option **-chksum** speichert Prüfsummenschlüsselwörter (CHECKSUM und DATASUM) im FITS-Header

Verweis: *setext*

savebmp

```
savebmp filename
```

Speichert das aktuelle Bild in Form einer Bitmap-Datei mit 8 Bit pro Kanal: **Dateiname**.bmp (BMP 24-bit)

savejpg

```
savejpg filename [quality]
```

Speichert das aktuelle Bild in eine JPG-Datei: **Dateiname**.jpg.

Die Qualität der Komprimierung kann mit dem optionalen Wert **quality** eingestellt werden, wobei 100 die beste und Standardeinstellung ist, während ein niedrigerer Wert die Komprimierung erhöht

savejxl

```
savejxl filename [-effort=] [-quality=] [-8bit]
```

Speichert das aktuelle Bild in eine JPG-XL-Datei: **filename**.jxl.

Alle anderen Argumente sind optional. Die Qualitätseinstellung drückt einen maximal zulässigen Unterschied zwischen dem Original und dem komprimierten Bild aus: Das Argument **-quality=** kann angegeben werden und muss als Fließkommazahl zwischen 0,0 und 10,0 angegeben werden. Eine höhere Qualität bedeutet eine bessere Qualität, aber auch eine größere Dateigröße. Qualität = 10,0 ist mathematisch verlustfrei, Qualität = 9,0 ist visuell verlustfrei und Qualität = 0 ist visuell schlecht, ergibt aber sehr kleine Dateigrößen. Der Standardwert ist 9,0; typische Werte reichen von 7,0 bis 10,0. Der Komprimierungsaufwand kann mit dem optionalen Wert **-effort=** eingestellt werden, wobei 9 den größten Aufwand bedeutet, aber sehr langsam ist, während ein niedrigerer Wert das Komprimierungsverhältnis erhöht. Werte über 7 werden nicht empfohlen, da sie sehr langsam sein können und nur wenig oder gar keinen Nutzen für die Dateigröße bringen; tatsächlich kann effort = 9 manchmal größere Dateien erzeugen. Wenn dieses Argument weggelassen wird, wird der Standardwert 7 verwendet. Die Option **-8bit** kann angegeben werden, um die Ausgabe mit 8 Bit pro Pixel zu erzwingen

savepng

```
savepng filename
```

Speichert das aktuelle Bild in eine PNG-Datei: **filename**.png, mit 16 Bit pro Kanal, wenn das geladene Bild 16 oder 32 Bit hat, und 8 Bit pro Kanal, wenn das geladene Bild 8 Bit hat

savepnm

```
savepnm filename
```

Speichert das aktuelle Bild in Form eines NetPBM-Dateiformats mit 16 Bit pro Kanal.

Die Erweiterung der Ausgabe ist **Dateiname**.ppm für RGB-Bilder und **Dateiname**.pgm für Graustufenbilder

savetif

```
savetif filename [-astro] [-deflate]
```

Speichert das aktuelle Bild in Form einer unkomprimierten TIFF-Datei mit 16 Bit pro Kanal: **filename**.tif. Die Option **-astro** ermöglicht das Speichern im Astro-Tiff-Format, während **-deflate** die Kompression aktiviert

Siehe auch SAVETIF32 und SAVETIF8

savetif32

```
savetif32 filename [-astro] [-deflate]
```

Gleicher Befehl wie SAVETIF, aber die Ausgabedatei wird mit 32 Bit pro Kanal gespeichert: **filename**.tif. Die Option **-astro** ermöglicht das Speichern im Astro-Tiff-Format, während **-deflate** die Kompression aktiviert

Verweis: *savetif*

savetif8

```
savetif8 filename [-astro] [-deflate]
```

Gleicher Befehl wie SAVETIF, aber die Ausgabedatei wird mit 8 Bit pro Kanal gespeichert: **filename**.tif. Die Option **-astro** ermöglicht das Speichern im Astro-Tiff-Format, während **-deflate** die Kompression aktiviert

Verweis: *savetif*

sb

```
sb [-loadpsf=] [-alpha=] [-iters=]
```

Stellt ein Bild nach der Split-Bregman-Methode wieder her.

Optional kann ein PSF mit dem Argument **-loadpsf=filename** geladen werden.

Die Anzahl der Iterationen wird durch **-iters** angegeben (Standard ist 1).

Der Regularisierungsfaktor **-alpha=** gibt die Stärke der Regularisierung an (kleinerer Wert = stärkere Regularisierung, Standardwert = 3000)

Verweis: *psf*

auswählen

```
select sequencename from to
```

Dieser Befehl ermöglicht eine einfache Auswahl von vielen Bildern in der geladenen Sequenz (von **from** bis **to** eingeschlossen). Dies ist eine Auswahl für eine spätere Verarbeitung.
Siehe auch UNSELECT

Verweise: *unselect*

Beispiele:

select . 0 0
wählt das Erste der aktuell geladenen Sequenz aus

select sequencename 1000 1200
Wählt 201 Bilder der Sequenz sequencename aus, beginnend mit Bild 1000

Die zweite Zahl kann größer als die Anzahl der Bilder in der Sequenz sein um bis zum Ende zu gehen.

seqapplyreg

```
seqapplyreg sequencename [-prefix=] [-scale=] [-layer=] [-framing=]
seqapplyreg sequencename ... [-interp=] [-noclamp]
seqapplyreg sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
seqapplyreg sequencename ... [-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-
↪filter-round=value[%|k]] [-filter-bkg=value[%|k]] [-filter-nbstars=value[%|k]] [-
↪filter-quality=value[%|k]] [-filter-incl[uded]]
```

Wendet geometrische Transformationen auf Bilder der im Argument angegebenen Sequenz an, so dass sie dem

Referenzbild überlagert werden können, wobei zuvor berechnete Registrierungsdaten verwendet werden (siehe REGISTER).

Der Name der Ausgabesequenz beginnt mit dem Präfix **"r_"**, sofern mit der Option **-prefix=** nichts anderes angegeben wurde.

Bei RGB-Bildern wird die Registrierung auf der ersten Ebene durchgeführt, für die Daten vorhanden sind, es sei denn, die Option **-layer=** (0, 1 oder 2 für R, G bzw. B) wurde angegeben.

Die ausgegebenen Bilder können durch Übergabe eines Wertes für das Argument **-scale=** (Skalierung, zwischen 0,1 und 3) vergrößert oder verkleinert werden.

Die automatische Beschneidung der Ausgabesequenz kann mit dem Schlüsselwort **-framing=**, gefolgt von einer der Methoden aus der Liste { current | min | max | cog } festgelegt werden:

-framing=max (Begrenzungsrahmen) projiziert jedes Bild und berechnet seine Verschiebung in Bezug auf das Referenzbild. Die resultierende Sequenz kann dann mit der Option **-maximize** des STACK-Befehls gestackt werden, wodurch das vollständige Bild erstellt wird, das alle Bilder der Sequenz umfasst.

-framing=min (gemeinsamer Bereich) beschneidet jedes Bild auf den Bereich, den es mit allen Bildern der Sequenz gemeinsam hat.

-framing=cog bestimmt die beste Position für den Bildausschnitt als Schwerpunkt (cog) aller Bilder.

Optionen zur Bildtransformation:

Standardmäßig werden Transformationen angewendet, um die Bilder mittels Interpolation zu registrieren.

Die Methode der Pixelinterpolation kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Wenn **none** übergeben wird, wird die Transformation erzwungen und eine pixelweise Verschiebung ohne Interpolation auf jedes Bild angewendet.

Die bikubische und die Lanczos4-Interpolationsmethode sind standardmäßig haltend, um Artefakte zu vermeiden, das kann aber mit dem Argument **-noclamp** deaktiviert werden.

Optionen zum Drizzeln:

Andernfalls können die Bilder mit dem HST-Drizzle-Algorithmus exportiert werden, indem das Argument **-drizzle** übergeben wird, das die folgenden zusätzlichen Optionen annehmen kann:

-pixfrac= legt den Pixelanteil fest (Standard = 1,0).

Das Argument **-kernel=** legt den Drizzle-Kernel fest und muss von einem der folgenden Argumente gefolgt werden: **point**, **turbo**, **square**, **gaussian**, **lanczos2** oder **lanczos3**. Der Standardwert ist **square**.

Das Argument **-flat=** gibt ein Master-Flat an, um die gedrizzelten Eingabepixel zu gewichten (Standard ist kein Flat).

Bilder herausfiltern:

Die zu registrierenden Bilder können auf der Grundlage einiger Filter ausgewählt werden (z. B. derjenigen mit der besten FWHM), mit einigen der **-filter-*** Optionen.

Links: *register*, *stack*

Einige davon werden gefiltert, ohne dass eine bestimmte Reihenfolge oder Anzahl festgelegt wird:

```
[-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-filter-round=value[%|k]] [-filter-
↪bkg=value[%|k]]
[-filter-nbstars=value[%|k]] [-filter-quality=value[%|k]] [-filter-incl[uded]]
```

Die besten Bilder der Sequenz können mit Hilfe der Filterargumente gestackt werden. Jedes dieser Argumente kann schlechte Bilder auf der Grundlage einer Eigenschaft, die ihrem Namen entpricht, aus den Registrierungsdaten mit einem der drei Typen von Argumentwerten entfernen:

- a numeric value for the worse image to keep depending on the type of data used (between 0 and 1 for roundness and quality, absolute values otherwise),

- a percentage of best images to keep if the number is followed by a % sign,

- or a k value for the k.sigma of the worse image to keep if the number is followed by a k sign.

Es ist auch möglich, manuell ausgewählte Bilder zu verwenden, entweder zuvor über die grafische Benutzeroberfläche oder mit den Befehlen select oder unselect, unter Verwendung des Arguments **-filter-included**.

seqccm

```
seqccm sequencename [-prefix=]
```

Gleicher Befehl wie CCM, aber für die Sequenz **sequencename**. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Name der Ausgabesequenz beginnt mit dem Präfix "ccm", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

Links: *ccm*

seqclean

```
seqclean sequencename [-reg] [-stat] [-sel]
```

Dieser Befehl löscht Auswahl-, Registrierungs- und/oder Statistikdaten, die für die Sequenz **sequencename** gespeichert wurden.

Mit den Optionen **-reg**, **-stat** und **-sel** können Sie festlegen, dass nur die Registrierung, die Statistik und/oder die Auswahl gelöscht wird. Wenn keine Option angegeben wird, werden alle gelöscht

seqcosme

```
seqcosme sequencename [filename].lst [-prefix=]
```

Gleicher Befehl wie COSME, aber für die Sequenz **sequencename**. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Name der Ausgabesequenz beginnt mit dem Präfix "cosme_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

Verweis: *cosme*

seqcosme_cfa

```
seqcosme_cfa sequencename [filename].lst [-prefix=]
```

Derselbe Befehl wie COSME_CFA, aber für die Sequenz **sequencename**. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Name der Ausgabesequenz beginnt mit dem Präfix "cosme_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

Verweis: *cosme_cfa*

seqcrop

```
seqcrop sequencename x y width height [-prefix=]
```

Beschneidet die im Argument **sequencename** angegebene Sequenz. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Ausschnitt wird durch die obere linke Eckposition **x** und **y** und die Auswahl **width** und **height** bestimmt, wie bei CROP.
Der Name der Ausgabesequenz beginnt mit dem Präfix "cropped_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweise: *crop*

seqextract_Green

```
seqextract_Green sequencename [-prefix=]
```

Gleicher Befehl wie EXTRACT_GREEN, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "Green_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

seqextract_Ha

```
seqextract_Ha sequencename [-prefix=] [-upscale]
```

Gleicher Befehl wie EXTRACT_HA, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "Ha_", sofern nicht mit der Option **-prefix=** etwas anderes angegeben wurde

seqextract_HaOIII

```
seqextract_HaOIII sequencename [-resample=]
```

Gleicher Befehl wie EXTRACT_HAOIII, aber für die Sequenz **sequencename**.

Die Namen der Ausgabesequenzen beginnen mit den Präfixen "Ha_" und "OIII_"

seqfind_cosme

```
seqfind_cosme sequencename cold_sigma hot_sigma [-prefix=]
```

Gleicher Befehl wie FIND_COSME, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "cc_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweis: *find_cosme*

seqfind_cosme_cfa

```
seqfind_cosme_cfa sequencename cold_sigma hot_sigma [-prefix=]
```

Gleicher Befehl wie FIND_COSME_CFA, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "cc_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweis: *find_cosme_cfa*

seqfindstar

```
seqfindstar sequencename [-layer=] [-maxstars=]
```

Gleicher Befehl wie FINDSTAR, aber für die Sequenz **sequencename**.

Die Option **-out=** ist für diesen Prozess nicht verfügbar, da alle Sternlistendateien mit dem Standardnamen *seqname_seqnb.lst* gespeichert werden

Verweis: *findstar*

seqfixbanding

```
seqfixbanding sequencename amount sigma [-prefix=] [-vertical]
```

Gleicher Befehl wie FIXBANDING, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "unband_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweis: *fixbanding*

seqght

```
seqght sequence -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie GHT, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *ght*

seqheader

```
seqheader sequencename keyword [keyword2 ...] [-sel] [-out=file.csv]
```

Gibt den FITS-Headerwert, der den angegebenen Schlüsseln für alle Bilder in der Sequenz entspricht aus. Sie können mehrere Schlüssel in einer Zeile schreiben, getrennt durch ein Leerzeichen. Mit der Option **-out=**, gefolgt von einem Dateinamen, können Sie die Ausgabe in einer CSV-Datei drucken. Mit der Option **-sel** wird die Ausgabe auf die in der Sequenz ausgewählten Bilder beschränkt.

seqinvght

```
seqinvght sequence -D= [-B=] [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie INVGHT, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *invght*

seqinvmodasinh

```
seqinvmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie INVMODASINH, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *invmodasinh*

seqlinstretch

```
seqlinstretch sequence -BP= [channels] [-sat] [-prefix=]
```

Gleicher Befehl wie LINSTRETCH, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *linstretch*

seqmerge_cfa

```
seqmerge_cfa sequencename0 sequencename1 sequencename2 sequencename3 bayerpattern [-
↪prefixout=]
```

Fügt 4 Bildsequenzen zusammen, um das Bayer-Muster neu zu kombinieren. Die Sequenzen werden in den Argumenten **sequencename0**, **sequencename1**, **sequencename2** und **sequencename3** angegeben.

Das zu rekonstruierende Bayer-Muster muss als zweites Argument als eines der folgenden angegeben werden: RGGB, BGGR, GBRG oder GRBG (die Reihenfolge der Bayer-Kanäle muss mit der Reihenfolge der angegebenen Sequenzen übereinstimmen).

Hinweis: Alle 4 Eingabesequenzen **müssen** vorhanden sein und dieselben Abmessungen, Bittiefe und Bildanzahl haben.

Der Name der Ausgabesequenz beginnt mit dem Präfix "mCFA_" und einer Zahl, sofern mit der Option **-prefixout=** nichts anderes angegeben wurde

seqmodasinh

```
seqmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Gleicher Befehl wie MODASINH, aber eine Sequenz muss als erstes Argument angegeben werden. Zusätzlich kann das optionale Argument **-prefix=** verwendet werden, um ein benutzerdefiniertes Präfix zu setzen

Verweise: *modasinh*

seqmtf

```
seqmtf sequencename low mid high [channels] [-prefix=]
```

Derselbe Befehl wie MTF, aber für die Sequenz **sequencename**.

Der Name der Ausgabesequenz beginnt mit dem Präfix "mtf_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

Verweis: *mtf*

seqprofile

```
seqprofile sequence -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] [-savedat] [-layer=] [-
↪width=] [-spacing=] [ {-xaxis=wavelength | -xaxis=wavenumber } ] [{-wavenumber1= | -
↪wavelength1=} -wn1at=x,y {-wavenumber2= | -wavelength2=} -wn2at=x,y] ["-title=My Plot"]
```

Erstellt ein Intensitätsprofildiagramm zwischen zwei Punkten in jedem Bild der Sequenz. Nach dem obligatorischen ersten Argument, das die zu verarbeitende Sequenz angibt, sind die anderen Argumente die gleichen wie für den Befehl **profile**. Wenn eine Sequenz verarbeitet wird und gewünscht wird, dass die aktuelle Bildnummer und die Gesamtzahl der Bilder im Format „Meine Sequenz (1 / 5)“ angezeigt werden, sollte der angegebene Titel mit () enden (z. B. „Meine Sequenz ()“ und die Nummern werden automatisch ausgefüllt)

seqpsf

```
seqpsf sequencename [channel] [{ -at=x,y | -wcs=ra,dec }] [-followstar]
```

Derselbe Befehl wie PSF, wird aber auf Sequenzen angewendet. Dieser Befehl ähnelt der Ein-Stern-Registrierung, mit dem Unterschied, dass die Ergebnisse für die photometrische Analyse verwendet werden können, anstatt die Bilder auszurichten, und dass die Koordinaten des Sterns über Optionen angegeben werden können.

Dieser Befehl wird intern vom Menü aufgerufen, das beim Rechtsklick auf das Bild mit dem PSF für den Sequenzeintrag angezeigt wird. Wenn bereits Registrierungsdaten für die Sequenz vorhanden sind, können diese verwendet werden, um das Suchfenster in jedem Bild zu verschieben. Wenn keine Registrierungsdaten vorhanden sind und wenn es erhebliche Verschiebungen zwischen den Bildern in der Sequenz gibt, können die Standardeinstellungen keine Sterne in der Ausgangsposition des Suchbereichs finden.

The follow star option can then be activated with the argument **-followstar**.

Die Ergebnisse werden auf der Registerkarte "Grafik" angezeigt, von der aus sie auch in eine CSV-Datei (Comma-Separated Values) zur externen Analyse exportiert werden können.

Bei der Erstellung einer Lichtkurve wird der erste Stern, für den seqpsf ausgeführt wurde und der in der Anzeige mit "V" markiert ist, als der veränderliche Stern betrachtet. Alle anderen werden gemittelt, um eine Referenzlichtkurve zu erstellen, die von der Lichtkurve des veränderlichen Sterns abgezogen wird.

Derzeit gibt der Befehl im Headless-Modus einige analysierte Daten in der Konsole aus. Mit einem weiteren Befehl können mehrere Sterne analysiert und als Lichtkurve dargestellt werden: LIGHT_CURVE. Argumente sind obligatorisch: Der Name der Sequenz muss angegeben werden („.“ kann verwendet werden, um die aktuell geladene Sequenz anzugeben), und im Headless-Modus müssen die Koordinaten des Sterns angegeben werden, wobei mit -at= Koordinaten in Pixeln für den Zielstern und mit -wcs= J2000-Äquatorialkoordinaten angegeben werden können.

Verweise: *psf*, *light_curve*

seqplatesolve

```
seqplatesolve sequencename [image_center_coords] [-focal=] [-pixelsize=]
seqplatesolve sequencename ... [-downscale] [-order=] [-radius=] [-force] [-noreg] [-
↪disto=]
seqplatesolve sequencename ... [-limitmag=[+-]] [-catalog=] [-nocrop] [-nocache]
seqplatesolve sequencename ... [-localasnet [-blindpos] [-blindres]]
```

Astrometrische Lösung einer Sequenz. Wenn die Eingabesequenz SER ist, wird eine neue Sequenz mit dem Präfix „ps_“ erstellt. Andernfalls werden die Bildkopfzeilen aktualisiert. Im Fall von SER ist die Bereitstellung der Metadaten obligatorisch und die Ausgabesequenz erfolgt im FITS-Cube-Format, da SER keine WCS-Daten speichern kann.

Wenn WCS oder andere Metadaten fehlerhaft sind oder fehlen, müssen Argumente angegeben werden:

Die ungefähren Koordinaten des Bildmittelpunkts können in Dezimalgraden oder Grad/Stunde-Minute-Sekunde-Werten (J2000 mit Doppelpunkten) angegeben werden, wobei die Werte für Rektaszension und Deklination durch ein Komma oder ein Leerzeichen getrennt sind (für astrometry.net nicht obligatorisch).

Brennweite und Pixelgröße können mit **-focal=** (in mm) und **-pixelsize=** (in µm) übergeben werden, wobei die Werte aus Bildern und Einstellungen überschrieben werden. Siehe auch Optionen zum blinden Lösen mit lokalem Astrometry.net

Für eine schnellere Erkennung von Sternen in großen Bildern ist eine Verkleinerung des Bildes mit **-downscale** möglich.

Bei der Lösung können Verzerrungen nach der SIP-Konvention mit Polynomen bis zur Ordnung 5 berücksichtigt

werden. Standardmäßig wird die kubische Verzerrung aus den Astrometrie-Einstellungen genutzt. Dies kann mit der Option **-order=** geändert werden, die einen Wert zwischen 1 und 5 angibt.

Wenn Sie lokale Siril-Solver-Kataloge oder lokales Astrometry.net verwenden und die anfängliche Lösung nicht erfolgreich ist, sucht der Solver nach einer Lösung innerhalb eines Radiuskegels, der mit der Option **-radius=** angegeben wird. Wenn kein Wert übergeben wird, wird der Suchradius aus den Astrometrieeinstellungen übernommen. Die Siril-Nähesuche kann durch Übergeben eines Werts von 0 deaktiviert werden. (kann für Astrometry.net nicht deaktiviert werden).

Bereits gelöste Bilder werden standardmäßig übersprungen. Dies kann durch Übergabe der Option **-force** deaktiviert werden.

Mit diesem Befehl werden die Registrierungsdaten aktualisiert, sofern nicht die Option **-noreg** übergeben wird.

Mit der Option **-disto=** können Sie die aktuelle astrometrische Lösung als Verzerrungsdatei speichern.

Astrometrische Lösungen für Bilder Bilder können entweder mit Siril unter Verwendung eines Sternkatalogs und des globalen Registrierungsalgorithmus oder mit dem Befehl solve-field vom lokalen astrometry.net (aktiviert mit **-localasnet**) ermittelt werden.

Optionen des Siril Lösers:

Die Grenzgröße der Sterne, die für die Plattenlösung verwendet werden, wird automatisch aus der Größe des Sichtfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.

Die Wahl des Sternkatalogs erfolgt automatisch, es sei denn, die Option **-catalog=** wird übergeben: Wenn lokale Kataloge installiert sind, werden sie verwendet, andernfalls erfolgt die Wahl auf der Grundlage des Sichtfelds und der Grenzgröße. Wenn die Option übergeben wird, erzwingt sie die Verwendung des im Argument angegebenen Remote-Katalogs, mit den möglichen Werten: tycho2, nomad, gaia, ppmxl, brightstars, apass.

Wenn das berechnete Bildfeld größer als 5 Grad ist, wird die Sternerkennung auf eine beschnittenen Bereich um das Bildzentrum begrenzt, falls nicht die Option **-nocrop** verwendet wird.

Bei der Verwendung von Online-Katalogen wird eine einzige Katalogextraktion für die gesamte Sequenz durchgeführt. Bei starker Drift oder unterschiedlicherm Sampling gelingt dies möglicherweise nicht für alle Bilder. Dies kann durch Übergabe des Arguments **-nocache** deaktiviert werden. In diesem Fall werden Metadaten von jedem Bild verwendet (mit Ausnahme der erzwungenen Werte wie Mittelpunktskoordinaten, Pixelgröße und/oder Brennweite).

Optionen des Astrometry.net Lösers:

Die Übergabe der Optionen **-blindpos** und/oder **-blindres** ermöglicht die blinde Lösung nach Position bzw. nach Sampling. Sie können diese verwenden, wenn Sie ein Bild mit einem völlig unbekannten Bildausschnitt und einem unbekannten Sampling lösen

seqresample

```
seqresample sequencename { -scale= | -width= | -height= } [-interp=] [-prefix=]
```

Skaliert die im Argument **sequencename** angegebene Sequenz. Nur ausgewählte Bilder in der Sequenz werden verarbeitet.

Der Skalierungsfaktor wird entweder durch das Argument **-scale=** oder durch Festlegen der Ausgabebreite, -höhe oder -maximaldimension mit den Optionen **-width=**, **-height=** oder **-maxdim=** angegeben.

Eine Interpolationsmethode kann mit dem Argument **-interp=** angegeben werden, gefolgt von einer der Methoden aus der Liste **ne**[arest] (nächster), **cu**[bic] (kubisch), **la**[nczos4], **li**[near], **ar**[ea] (Bereich)}. Bei der kubischen und der Lanczos-Interpolation wird die Klammerung angewendet.

Der Name der Ausgabesequenz beginnt mit dem Präfix "scaled_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde

seqrl

```
seqrl sequencename [-loadpsf=] [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Dasselbe wir der Befehl RL, gilt jedoch für eine Sequenz, die als erstes Argument angegeben werden muss

Verweise: *rl*

seqsb

```
seqsb sequencename [-loadpsf=] [-alpha=] [-iters=]
```

Dasselbe wir der Befehl SB, gilt jedoch für eine Sequenz, die als erstes Argument angegeben werden muss

Verweise: *sb*

seqsetmag

```
seqsetmag magnitude
```

Wie der Befehl SETMAG, aber für die geladene Sequenz.

Dieser Befehl ist nur gültig, nachdem Sie SEQPSF oder sein grafisches Gegenstück ausgeführt haben (wählen Sie den Bereich um einen Stern aus und starten Sie die PSF-Analyse für die Sequenz; sie wird in der Registerkarte "Grafischer Plot" angezeigt).
Dieser Befehl hat das gleiche Ziel wie SETMAG, berechnet aber die Referenzgröße für jedes Bild der Sequenz, in der der Referenzstern gefunden wurde, neu.

Beim Ausführen des Befehls wird der zuletzt analysierte Stern als Referenzstern betrachtet. Die Anzeige des Magnituden-Diagramms vor der Eingabe des Befehls erleichtert das Verständnis.
Um den Referenzstern und den Magnituden-Offset zurückzusetzen, siehe SEQUNSETMAG

Verweise: *setmag*, *seqpsf*, *psf*, *sequnsetmag*

seqsplit_cfa

```
seqsplit_cfa sequencename [-prefix=]
```

Gleicher Befehl wie SPLIT_CFA, aber für die Sequenz **sequencename**.

Die Namen der Ausgabesequenzen beginnen mit dem Präfix "CFA_" und einer Zahl, sofern mit der Option **-prefix=** nichts anderes angegeben wird.
Einschränkung: Die Sequenz gibt immer eine Folge von FITS-Dateien aus, unabhängig von der Art der Eingabesequenz

Verweis: *split_cfa*

seqstarnet

```
seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Dieser Befehl ruft Starnet++ auf, um Sterne aus der Sequenz **sequencename** zu entfernen. Siehe STARNET

Verweis: *starnet*

seqstat

```
seqstat sequencename output_file [option] [-cfa]
```

Gleicher Befehl wie STAT für die Sequenz **sequencename**.

Die Daten werden als CSV-Datei **output_file** gespeichert.

Der optionale Parameter bestimmt die Anzahl der berechneten statistischen Werte: **basic**, **main** (Standard) oder **full** (detaillierter, aber länger in der Berechnung).
\t**basic** umfasst Mittelwert, Median, Sigma, bgnoise (Hintergrundrauschen), min und max
\t**main**umfasst die Basisdaten mit dem Zusatz von avgDev (Standardabweichung), MAD und der Quadratwurzel von BWMV
\t**full** umfasst "main" mit dem Zusatz von Standort und Maßstab.

Wenn **-cfa** übergeben wird und die Bilder CFA sind, werden Statistiken über Extraktionen pro Filter erstellt

Verweis: *stat*

seqsubsky

```
seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-
↪smooth=0.5] [-prefix=]
```

Gleicher Befehl wie SUBSKY, aber für die Sequenz **sequencename**.
Dithering, required for low dynamic gradients, can be disabled with **-nodither**. Note that the **-existing** option is not available for sequence background removal, as the frames of a sequence are not necessarily always aligned.

Der Name der Ausgabesequenz beginnt mit dem Präfix "bkg_", sofern mit der Option **-prefix=** nichts anderes angegeben wurde. Nur ausgewählte Bilder der Sequenz werden verarbeitet

Verweis: *subsky*

seqtilt

```
seqtilt sequencename
```

Derselbe Befehl wie TILT, aber für die Sequenz **sequencename**. Ergibt generell bessere Resultate

Verweit: *tilt*

sequnsetmag

```
sequnsetmag
```

Setzt die Magnitudenkalibrierung und den Referenzstern für die Sequenz zurück. Siehe SEQSETMAG

Verweis: *seqsetmag*

sequpdate_key

```
sequpdate_key sequencename key value [keycomment]
sequpdate_key sequencename -delete key
sequpdate_key sequencename -modify key newkey
sequpdate_key sequencename -comment comment
```

Gleicher Befehl wie UPDATE_KEY, aber für die Sequenz **Sequenzname**. Dieser Befehl funktioniert jedoch nicht bei SER-Sequenzen

Links: *update_key*

seqwiener

```
wiener sequencename [-loadpsf=] [-alpha=]
```

Dasselbe wie der Befehl **WIENER**, aber für eine Sequenz, die als erstes Argument angegeben werden muss

Verweise: *wiener*

set

```
set { -import=inifilepath | variable=value }
```

Aktualisieren eines Einstellungswerts unter Verwendung seines Variablennamens mit dem angegebenen Wert oder einer Reihe von Werten unter Verwendung einer vorhandenen ini-Datei mit der Option **-import=**.
Siehe GET, um Werte oder die Liste der Variablen zu erhalten

Verweise: *get*

set16bits

```
set16bits
```

Deaktiviert die Speicherung von Bildern mit 32 Bit pro Kanal bei der Verarbeitung. Es werden stattdessen 16 Bits verwendet

set32bits

```
set32bits
```

Ermöglicht das Speichern von Bildern mit 32 Bit pro Kanal bei der Verarbeitung

setcompress

```
setcompress 0/1 [-type=] [q]
```

Legt fest, ob Bilder komprimiert werden oder nicht.

0 bedeutet keine Kompression, während **1** die Kompression aktiviert.
Wenn die Komprimierung aktiviert ist, muss der Typ ausdrücklich in der Option **-type=** angegeben werden ("rice", "gzip1", "gzip2").
Im Zusammenhang mit der Komprimierung muss der Quantisierungswert im Bereich [0, 256] liegen.

Beispiel: "setcompress 1 -type=rice 16" stellt die Rice-Kompression mit einer Quantisierung von 16 ein

setcpu

```
setcpu number
```

Legt die Anzahl der für die Berechnung verwendeten Verarbeitungsthreads fest.

Kann so hoch sein wie die Anzahl der auf dem System vorhandenen virtuellen Threads, d. h. die Anzahl der CPU-Kerne oder das Doppelte dieser Anzahl, wenn Hyperthreading (Intel HT) verfügbar ist. Der Standardwert ist die maximale Anzahl von Threads, die zur Verfügung stehen, so dass dieser Wert hauptsächlich zur Begrenzung der Rechenleistung verwendet werden sollte. Dieser Wert wird bei jedem Siril-Lauf zurückgesetzt. Siehe auch SETMEM

Verweise: *setmem*

setext

```
setext extension
```

Legt die verwendete und von Sequenzen erkannte Erweiterung fest.

Das Argument **Erweiterung** kann "fit", "fts" oder "fits" sein

setfindstar

```
setfindstar [reset] [-radius=] [-sigma=] [-roundness=] [-focal=] [-pixelsize=] [-
↪convergence=] [ [-gaussian] | [-moffat] ] [-minbeta=] [-relax=on|off] [-minA=] [-
↪maxA=] [-maxR=]
```

Definiert die Parameter für die Sternerkennung für die Befehle FINDSTAR und REGISTER.

Wird kein Parameter übergeben, werden die aktuellen Werte aufgelistet.
Die Übergabe von **reset** setzt alle Werte auf die Standardwerte zurück. Sie können dann immer noch Werte nach diesem Schlüsselwort übergeben.

Konfigurierbare Werte:

-radius= definiert den Radius des anfänglichen Suchfeldes und muss zwischen 3 und 50 liegen.
-sigma= definiert die Schwelle über dem Rauschen und muss größer oder gleich 0,05 sein.
-roundness= definiert die minimale Rundheit der Sterne und muss zwischen 0 und 0,95 liegen. **-maxR** erlaubt es, eine Obergrenze für die Rundheit festzulegen, um nur die Bereiche zu visualisieren, in denen die Sterne deutlich elongiert sind. Nicht ändern für die Registrierung.
-minA und **-maxA** definieren Grenzwerte für die minimale und maximale Amplitude der Sterne, die zwischen 0 und 1 normiert sind.
-focal= definiert die Brennweite des Teleskops.
-pixelsize= definiert die Pixelgröße des Sensors.

-gaussian und **-moffat** konfigurieren das zu verwendende Solver-Modell (Gaussian ist der Standard).
Wenn Moffat gewählt wird, definiert **-minbeta=** den Mindestwert von beta, für den Kandidatensterne akzeptiert werden, und muss größer oder gleich 0,0 und kleiner als 10,0 sein.
-convergence= definiert die Anzahl der Iterationen, die zur Anpassung der PSF durchgeführt werden, und sollte zwischen 1 und 3 (toleranter) eingestellt werden.
-relax= entspannt die Prüfungen, die bei Sternkandidaten durchgeführt werden, um festzustellen, ob sie Sterne sind oder nicht, damit Objekte, die nicht wie Sterne geformt sind, trotzdem akzeptiert werden (standardmäßig ausgeschaltet)

Verweis: *findstar*, *register*, *psf*

Der Schwellenwert für die Sternerkennung wird berechnet als Median des Bildes (der im Allgemeinen den Hintergrundpegel darstellt) plus k mal sigma, wobei sigma die Standardabweichung des Bildes ist (ein guter Hinweis auf die Rauschamplitude). Wenn Sie viele Sterne in Ihren Bildern und ein gutes Signal-Rausch-Verhältnis haben, kann es eine gute Idee sein, diesen Wert zu erhöhen, um die Erkennung zu beschleunigen und fehlerhafte erkannte Sterne zu vermeiden.

Es wird empfohlen, die für eine Sequenz verwendeten Werte mit der grafischen Benutzeroberfläche von Siril zu testen, die im dynamischen PSF-Werkzeugkasten des Analysemenüs verfügbar ist. Es kann die Qualität der Registrierung verbessern, wenn die Parameter erhöht werden, aber es ist auch wichtig, dass mehrere Dutzend Sterne in jedem Bild erkannt werden.

setmag

```
setmag magnitude
```

Kalibriert die Helligkeit durch Auswahl eines Sterns und Angabe der bekannten scheinbaren Magnitude.

Alle PSF-Berechnungen liefern anschließend die kalibrierte scheinbare Magnitude anstelle einer scheinbaren Helligkeit relativ zu ADU-Werten. Beachten Sie, dass der angegebene Wert mit der Helligkeit übereinstimmen muss, damit der Beobachtungsfilter sinnvoll ist.
Zum Zurücksetzen der Magnitudenkonstante siehe UNSETMAG

Verweis: *psf*, *unsetmag*

setmem

```
setmem ratio
```

Legt einen neuen Anteil des freien Speichers am Gesamtspeicher fest, der verwendet werden kann.

Der Wert **Ratio** sollte zwischen 0,05 und 2 liegen, je nach den anderen Aktivitäten des Computers. Ein höherer Anteil sollte es Siril ermöglichen, schneller zu Arbeiten, aber die Einstellung des Anteils des für das Stacken verwendeten Speichers über 1 erfordert die Verwendung von Festplattenspeicher als Auslagerungsspeicher, was sehr langsam ist und nicht empfohlen wird und manchmal gar nicht unterstützt wird und zu einem Systemabsturz führen kann. Eine feste Menge des RAMs kann auch in den allgemeinen Einstellungen mit SET anstelle eines Verhältnisses festgelegt werden

Verweise: *set*

setphot

```
setphot [-inner=20] [-outer=30] [-aperture=10] [-dyn_ratio=4.0] [-gain=2.3] [-min_val=0]␣
↪[-max_val=60000]
```

Liest oder setzt die Einstellungen für die Photometrie, die hauptsächlich von SEQPSF verwendet werden. Wenn Argumente angegeben werden, aktualisieren sie die Einstellungen. Keine der Einstellungen ist obligatorisch, es können alle optional angegeben werden, die Standardwerte sind in der Befehlssyntax angegeben. Am Ende des Befehls wird die aktive Konfiguration ausgedruckt.

Die Blendengröße ist dynamisch, es sei denn, sie wird erzwungen. Ist dies der Fall, wird der Wert **Blende** aus den Einstellungen verwendet. Wenn dynamisch, wird der Radius der Blende durch das angegebene dynamische Verhältnis („Radius/halber FWHM“) definiert.
Die zulässigen Werte für das Argument **-dyn_ratio** liegen im Bereich [1.0, 5.0]. Ein Wert außerhalb dieses Bereichs setzt die Blende automatisch auf den festen Wert **-aperture**.

Der Gainwert wird nur genutzt, wenn der nicht aus dem FITS-Header gelesen werden kann

Verweis: *seqpsf*

setref

```
setref sequencename image_number
```

Legt das Referenzbild der im ersten Argument angegebenen Sequenz fest. **image_number** ist die fortlaufende Nummer des Bildes in der Sequenz, nicht die Nummer im Dateinamen, beginnend bei 1

show

```
show [-clear] [{ -list=file.csv | [name] RA Dec }] [-nolog] [-notag]
```

Zeigt einen Punkt auf dem astrometrisch gelösten Bild unter Verwendung des temporären Katalogs der Benutzeranmerkungen, basierend auf seinen äquatorialen Koordinaten. Die Option **-clear** löscht diesen Katalog zuerst und kann allein verwendet werden.
Mehrere Punkte können über eine CSV-Datei mit der Option **-list=** übergeben werden, die mindestens Ra- und Dec-Spalten enthält. Wenn die übergebene Datei auch eine Spalte mit Namen enthält, werden die Namen als Tags im Bild verwendet und in der Konsole aufgeführt, es sei denn, sie werden mit den Optionen **-notag** und **-nolog** ausgeschaltet.

Diese Funktion ist nur über die GUI von Siril verfügbar

spcc

```
spcc [-limitmag=[+-]] [ { -monosensor= [ -rfilter= ] [-gfilter=] [-bfilter=] | -
↪oscsensor= [-oscfilter=] [-osclpf=] } ] [-whiteref=] [ -narrowband [-rwl=] [-gwl=] [-
↪bwl=] [-rbw=] [-gbw=] [-bbw=] ] [-bgtol=lower,upper] [ -atmos [-obsheight=] { [-
↪pressure=] | [-slp=] } ]
```

Führt die spektrophotometrische Farbkalibrierung auf dem geladenen, astrometrisch gelösten Bild durch.

Die Grenzgröße der Sterne wird automatisch aus der Größe des Bildfeldes berechnet, kann aber durch Übergabe eines +Offset- oder -Offset-Wertes an **-limitmag=** oder einfach durch einen absoluten positiven Wert für die Grenzgröße geändert werden.
The star catalog used for SPCC is always Gaia DR3: by default the local Gaia DR3 xp_sampled catalog will be used if available but this can be overridden with **-catalog={gaia | localgaia}**.

Die Namen der Sensoren und Filter können mit den folgenden Optionen angegeben werden: **-monosensor=**, **-rfilter=**, **-gfilter=**, **-bfilter=** oder **-oscsensor=**, **-oscfilter=**, **-osclpf=**; der Name der Weißreferenz kann mit der Option **-whiteref=** angegeben werden. In allen Fällen muss der Name genau so angegeben werden, wie er in den Kombinationsfeldern des SPCC-Tools steht. Beachten Sie, dass Sensor-, Filter- und Weißreferenznamen Leerzeichen enthalten können: Wenn Sie sie als Argumente für den Befehl **spcc** verwenden, muss das gesamte Argument in Anführungszeichen gesetzt werden, zum Beispiel "-whiteref=Average Spiral Galaxy".

Der Schmalbandmodus kann mit dem Argument **-narrowband** ausgewählt werden. In diesem Fall werden die vorherigen Filterargumente ignoriert, und die Wellenlängen und Bandbreiten der NB-Filter können mit **-rwl=**, **-rbw=**, **-gwl=**, **-gbw=**, **-bwl=** und **-bbw=** angegeben werden.

Wird eines der Spektraldaten-Argumente weggelassen, wird der zuvor verwendete Wert verwendet.

Die Toleranz für Außreißer der Hintergrundreferenz kann in Sigma-Einheiten mit **-bgtol=lower,upper** angegeben werden: die Standardwerte sind -2,8 und +2,0

Eine atmosphärische Korrektur kann durch Übergabe von **-atmos** vorgenommen werden. In diesem Fall gelten die folgenden optionalen Argumente: **-obsheight=** gibt die Höhe des Beobachters über dem Meeresspiegel in Metern an (Standard 10), **-pressure=** gibt den lokalen Luftdruck am Beobachtungsort in hPa an oder **-slp=** gibt den Luftdruck auf Meereshöhe in hPa an (der Standarddruck beträgt 1013,25 hPa auf Meereshöhe).

spcc_list

```
spcc_list { oscsensor | monosensor | redfilter | greenfilter | bluefilter | oscfilter |␣
↪osclpf | whiteref }
```

Zeigt eine Liste von SPCC-Namen an, die zur Definition von Sensoren, Filtern oder Weißreferenzen mit dem Befehl **spcc** verwendet werden können. Dieser Befehl erfordert ein Argument, um festzulegen, welche Liste gedruckt werden soll: Die Optionen sind **oscsensor**, **monosensor**, **redfilter**, **greenfilter**, **bluefilter**, **oscfilter**, **osclpf** oder **whiteref**.

Beachten Sie, dass Sensor-, Filter- und Weißreferenznamen Leerzeichen enthalten können: Wenn Sie sie als Argumente für den Befehl **spcc** verwenden, muss das gesamte Argument in Anführungszeichen gesetzt werden, zum Beispiel "-whiteref=Average Spiral Galaxy"

Verweise: *spcc*

split

```
split file1 file2 file3 [-hsl | -hsv | -lab]
```

Teilt ein Farbbild in drei verschiedene Dateien auf (eine für jede Farbe) und speichert sie in den Dateien **file1**.fit, **file2**.fit und **file3**.fit. Als letztes Argument kann optional angegeben werden, **-hsl**, **-hsv** oder **lab**, um eine HSL-, HSV- oder CieLAB-Extraktion durchzuführen. Wenn keine Option angegeben wird, erfolgt die Extraktion im RGB-Format, d.h. es wird keine Konvertierung durchgeführt

split_cfa

```
split_cfa
```

Teilt das geladene CFA-Bild in vier verschiedene Dateien (eine für jeden Kanal) und speichert sie in Dateien

stack

```
stack seqfilename
stack seqfilename { sum | min | max } [-output_norm] [-out=filename] [-maximize] [-
↪upscale] [-32b]
stack seqfilename { med | median } [-nonorm, -norm=] [-fastnorm] [-rgb_equal] [-output_
↪norm] [-out=filename] [-32b]
stack seqfilename { rej | mean } [rejection type] [sigma_low sigma_high]  [-rejmap[s]] [-
↪nonorm, -norm=] [-fastnorm] [-overlap_norm] [-weight={noise|wfwhm|nbstars|nbstack}] [-
↪feather=] [-rgb_equal] [-output_norm] [-out=filename] [-maximize] [-upscale] [-32b]
```

Stackt die Sequenz **sequencename** unter Verwendung von Optionen.

Stacking Methode:

Die erlaubten Typen sind: "sum", "max", "min", "med" (oder "median") und "rej" (oder "mean"). Wird kein anderes Argument als der Name der Sequenz angegeben, wird Summenstacking verwendet.

Stapeln mit Ausschluss (Rejection):

Die Typen **rej** oder **mean** erfordern die Verwendung zusätzlicher Argumente für den Pixelausschluss und entsprechende Werte. Der Ausschlusstyp ist einer von **n[one] | p[ercentile] | s[igma] | m[edian] | w[insorized] | l[inear] | g[eneralized] | [m]a[d]** für Percentile, Sigma, Median, Winsorized, Linear-Fit, Generalized Extreme Studentized Deviate Test oder k-MAD Clipping. Wird dies nicht angegeben, wird die Standardeinstellung (Winsorized) verwendet.

Die Parameter sigma low und sigma high des Ausschlusses sind obligatorisch solange nicht none ausgewählt wird.

Optional können Ausschlusskarten (Rejection maps) erzeugt werden die zeigen wo Pixel in einer (-rejmap) oder zwei (-rejmaps), für niedrige und hohe Grenzwerte, ausgeschlossen werden.

Normalisierung der Eingabebilder:

Für die Stackingmethoden **med** (oder **median**) und **rej** (oder **mean**) sind verschiedene Arten der Normalisierung zulässig: **-norm=add** für additiv, **-norm=mul** für multiplikativ. Die Optionen **-norm=addscale** und **-norm=mulscale** wenden die gleiche Normalisierung an, aber mit Skalierungsoperationen. **-nonorm** ist die Option zum Ausschalten der Normalisierung. Ansonsten wird standardmäßig die additive Methode mit Skalierung angewendet.

Die Option -fastnorm legt fest, dass schnellere Schätzungen für Ort und Maßstab als der Standard-IKSS verwendet werden sollen.

-overlap_norm berechnet, wenn übergeben, Normalisierungskoeffizienten für Bildüberlappungen statt für ganze Bilder (nur zulässig, wenn **-maximize** übergeben wird).

Weitere Optionen für das Stacking mit Ausschluss:

Mit der Option **-weight=** können die Bilder der Sequenzen gewichtet werden. Darauf folgt:

noise, um Frames mit geringerem Hintergrundrauschen größere Gewichte hinzuzufügen.

nbstack zum Gewichten von Eingabebildern basierend darauf, wie viele Bilder zu ihrer Erstellung verwendet wurden, nützlich für Live-Stacking.

nbstars oder **wfwhm**, um Eingabebilder basierend auf der Anzahl der Sterne oder wFWHM zu gewichten, die während des Registrierungsschritts berechnet wurden.

Die Option **-feather=** wendet auf alle Bildränder eine Überblendungsmaske über die im Argument angegebene Distanz (in Pixeln) an.

Ausgabe:

Der Name des Ergebnisbildes kann mit der Option -out= festgelegt werden. Andernfalls wird es als sequencename_stacked.fit benannt.

-output_norm wendet eine Normalisierung am Ende des Stackings an, um das Ergebnis im Bereich [0, 1] neu zu skalieren.

Die Option **-maximize** verwendet Registrierungsdaten aus der Sequenz, um ein gestacktes Bild zu erstellen, das alle Bilder der Sequenz umfasst (gilt für alle Methoden außer Median-Stacking).

Die Option **-upscale** skaliert die Sequenz vor dem Stacking mithilfe der Registrierungsdaten um den Faktor 2 hoch (gilt für alle Methoden außer Median-Stacking).

-rgb_equal verwendet die Normalisierung, um Farbbildhintergründe anzugleichen. Dies ist nützlich, wenn PCC/SPCC oder nicht verknüpftes AUTOSTRETCH nicht verwendet werden.

-32b überschreibt die in den Einstellungen festgelegte Bittiefe und speichert das gestackte Bild in 32-Bit.

Bilder herausfiltern:

Bilder, die gestapelt werden sollen, können auf der Grundlage einiger Filter, wie manuelle Auswahl oder beste FWHM, mit einigen der **-filter-** Optionen ausgewählt werden.

Links: *pcc*, *spcc*, *autostretch*

```
[-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-filter-round=value[%|k]] [-filter-
↪bkg=value[%|k]]
[-filter-nbstars=value[%|k]] [-filter-quality=value[%|k]] [-filter-incl[uded]]
```

Die besten Bilder der Sequenz können mit Hilfe der Filterargumente gestackt werden. Jedes dieser Argumente kann schlechte Bilder auf der Grundlage einer Eigenschaft, die ihrem Namen entpricht, aus den Registrierungsdaten mit einem der drei Typen von Argumentwerten entfernen:

- a numeric value for the worse image to keep depending on the type of data used (between 0 and 1 for roundness and quality, absolute values otherwise),

- a percentage of best images to keep if the number is followed by a % sign,

- or a k value for the k.sigma of the worse image to keep if the number is followed by a k sign.

Es ist auch möglich, manuell ausgewählte Bilder zu verwenden, entweder zuvor über die grafische Benutzeroberfläche oder mit den Befehlen select oder unselect, unter Verwendung des Arguments **-filter-included**.

stackall

```
stackall
stackall { sum | min | max } [-maximize] [-upscale] [-32b]
stackall { med | median } [-nonorm, norm=] [-32b]
stackall { rej | mean } [rejection type] [sigma_low sigma_high] [-nonorm, norm=] [-
↪overlap_norm] [-weight={noise|wfwhm|nbstars|nbstack}] [-feather=] [-rgb_equal] [-
↪out=filename] [-maximize] [-upscale] [-32b]
```

Öffnet alle Sequenzen im aktuellen Verzeichnis und stapelt sie mit dem optional angegebenen Stapeltyp und Filterung oder mit Summenstapelung. Siehe STACK-Befehl für die Beschreibung der Optionen

Verweis: *stack*

starnet

```
starnet [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Dieser Befehl ruft StarNet auf, um Sterne aus dem aktuellen Bild zu entfernen.

Voraussetzung: StarNet ist ein externes Programm, das nicht automatisch mit Siril installiert wird, und muss vor der ersten Verwendung dieses Befehls korrekt installiert werden, wobei der Pfad zu seinem Installationsverzeichnis in Einstellungen / Verschiedenes korrekt eingestellt sein muss. Das Verzeichnis muss die Kommandozeilenversion des Programms enthalten (nicht die GUI-Version, die für Windows-Benutzer existiert).

Das sternlose Bild wird bei Fertigstellung geladen, und ein Bild mit der Sternmaske wird im Arbeitsverzeichnis erstellt, sofern nicht der optionale Parameter **-nostarmask** angegeben wird.

Optional können dem Befehl auch Parameter übergeben werden:
- The option **-stretch** is for use with linear images and will apply a pre-stretch before running StarNet and the inverse stretch to the generated starless and starmask images.
- To improve star removal on images with very tight stars, the parameter **-upscale** may be provided. This will upsample the image by a factor of 2 prior to StarNet processing and rescale it to the original size afterwards, at the expense of more processing time.
- The optional parameter **-stride=value** may be provided, however the author of StarNet *strongly* recommends that the default stride of 256 be used

start_ls

```
start_ls [-dark=filename] [-flat=filename] [-rotate] [-32bits]
```

Initialisiert eine Livestacking-Sitzung unter Verwendung der optionalen Kalibrierungsdateien und wartet auf Eingabedateien, die durch den Befehl LIVESTACK bereitgestellt werden, bis STOP_LS aufgerufen wird. Standardmäßig wird die Registrierung nur mit Verschiebung und 16-Bit-Verarbeitung durchgeführt, da dies schneller ist. Dies kann mit **-rotate** und **-32bits** auf Rotation und 32 Bit geändert werden

Beachten Sie, dass die Live-Stacking-Befehle Siril in einen Zustand versetzen, in dem es nicht in der Lage ist, andere Befehle zu verarbeiten. Nach START_LS können nur noch LIVESTACK, STOP_LS und EXIT aufgerufen werden, bis STOP_LS aufgerufen wird, um Siril in den normalen, nicht-Live-Stacking-Zustand zurückzubringen

Verweise: *livestack*, *stop_ls*, *exit*

stat

```
stat [-cfa] [main]
```

Liefert Statistiken für das aktuelle Bild, standardmäßig die Basiswerte oder die Hauptliste, wenn **main** übergeben wird. Wenn eine Auswahl getroffen wird, werden die Statistiken innerhalb der Auswahl berechnet. Wenn **-cfa** übergeben wird und es sich um ein CFA-Bild handelt, werden die Statistiken für die einzelnen Farbkanäle erstellt

stop_ls

```
stop_ls
```

Beendet die Live-Stacking-Sitzung. Nur möglich nach START_LS

Verweis: *start_ls*

subsky

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-
↪existing]
```

Berechnet einen synthetischen Hintergrundgradienten unter Verwendung des Polynomfunktionsmodells des Grades **degree** oder des RBF-Modells (wenn stattdessen **-rbf** angegeben wird) und subtrahiert ihn vom Bild.

Die Anzahl der Stichproben pro horizontaler Linie und die Toleranz zum Ausschluss hellerer Bereiche können mit den optionalen Argumenten angepasst werden. Die Toleranz wird in MAD-Einheiten angegeben: Median + Toleranz * mad.

Dithering, das für geringe dynamische Gradienten erforderlich ist, kann mit **-dither** aktiviert werden.

For RBF, the additional smoothing parameter is also available. To use pre-existing background samples (e.g. if you have set background samples using a Python script) the **-existing** argument must be used

synthstar

```
synthstar
```

Korrigiert schlechte Sterne im geladenen Bild. Unabhängig davon, wie viel Koma, Nachführdrift oder andere Verzerrungen Ihre Sterne haben, wenn Sirils Sternsucherroutine sie erkennen kann, wird synthstar sie korrigieren. Wenn Sie besonders vorsichtig sein wollen, sollten Sie alle Sterne, die Sie korrigieren möchten, manuell auswählen. Dazu können Sie den Konsolenbefehl findstar oder den Dialog Dynamischer PSF verwenden. Wenn Sie die Sternsuche nicht ausgeführt haben, wird sie automatisch mit den Standardeinstellungen ausgeführt.

Die besten Ergebnisse erzielen Sie, wenn Sie Synthstar vor dem Strecken ausführen.

Das Ergebnis von synthstar ist eine vollständig korrigierte synthetische Sternmaske mit perfekt runden Stern-PSFs (Moffat- oder Gauß-Profile je nach Sternsättigung), die so berechnet werden, dass sie mit der Intensität, der FWHM, dem Farbton und der Sättigung übereinstimmen, die für jeden im Eingabebild erkannten Stern gemessen wurden. Diese kann dann mit dem sternlosen Bild rekombiniert werden, um ein Bild mit perfekten Sternen zu erzeugen.

Für diesen Befehl sind keine Parameter erforderlich

Verweis: *psf*

threshlo

```
threshlo level
```

Ersetzt Werte unter **level** durch **level** im geladenen Bild

threshhi

```
threshi level
```

Ersetzt Werte über **level** durch **level** im geladenen Bild

thresh

```
thresh lo hi
```

Ersetzt Werte unter **level** durch **level** im geladenen Bild

tilt

```
tilt [clear]
```

Berechnet die Verkippung des Sensors als FWHM-Differenz zwischen dem besten und dem schlechtesten abgeschnittenen Mittelwert der Ecken. Die Option **clear** erlaubt es, die Darstellung zu löschen

trixel

```
trixel [-p]
```

Für Entwickler.

Listet ohne Argument alle im astrometrisch gelösten Bild sichtbaren Trixel der Ebene 3 auf. Die Sterne von jedem Trixel können dann mit dem Befehl CONESEARCH unter Verwendung von **-trix=** gefolgt von einer sichtbaren Trixel-Nummer angezeigt werden

Mit dem Argument **-p** werden alle validen Sterne aller 512 Level3-Trixel in eine Datei "trixels.csv" geschrieben

Links: *conesearch*

unclipstars

```
unclipstars
```

Reprofiliert übersättigte Sterne, um sie zu entsättigen, und skaliert die Ausgabe so, dass alle Pixelwerte <= 1,0 sind

unpurple

```
unpurple [-starmask] [-blue=value] [-thresh=value]
```

Wendet einen kosmetischen Filter an, um die Auswirkungen violetter Farbsäume auf Sternen zu verringern.

Wenn der Parameter **-starmask** angegeben ist, wird eine Sternmaske verwendet, um die zu beeinflussenden Bildbereiche zu identifizieren. Wenn bereits eine dynamische PSF ausgeführt wurde, wird diese für die Sternmaske verwendet, andernfalls wird automatisch eine erstellt. Der Parameter **-mod=** sollte einen Wert um 0,14 erhalten, um die Menge an Lila zu reduzieren. **-thresh=** gibt den Größenmodifikator für jeden Stern in der Sternmaske an und sollte groß genug sein, damit die Sterne vollständig verarbeitet werden, ohne dass lila Ränder verbleiben. Der Wert sollte zwischen 0 und 1 liegen, normalerweise um 0,5.

Wenn der Parameter **-starmask** nicht angegeben ist, wird die Violettreduzierung auf das gesamte Bild für alle violetten Pixel angewendet, deren Luminanzwert höher ist als der angegebene **-thresh=**. In diesem Fall sollte der **-thresh=**-Wert angemessen niedrig sein. Dieser Modus ist nützlich für Sternmasken oder andere Bilder ohne Nebel oder Galaxien

Verweis: *psf*

unselect

```
unselect sequencename from to
```

Ermöglicht die einfache Abwahl von vielen Bildern in der geladenen Sequenz (von **from** bis **to** eingeschlossen). Siehe SELECT

Verweis: *select*

unsetmag

```
unsetmag
```

Setze die Magnituden-Kalibrierung auf 0 zurück. Vgl. SETMAG

Verweis: *setmag*

unsharp

```
unsharp sigma multi
```

Wendet eine Unschärfemaske an, d. h. ein Gaußfilter mit Sigma **sigma** und einer Überblendung mit dem Parameter **amount**, die wie folgt verwendet wird: out = in * (1 + amount) + filtered * (-amount).

Siehe auch GAUSS, dasselbe ohne Überblendung

Verweise: *gauss*

update_key

```
update_key key value [keycomment]
update_key -delete key
update_key -modify key newkey
update_key -comment comment
```

Aktualisiert das FITS-Schlüsselwort. Bitte beachten Sie, dass die Gültigkeit von **value** nicht überprüft wird. Diese Überprüfung liegt in der Verantwortung des Benutzers. Es ist auch möglich, einen Schlüssel mit der Option **-delete** vor dem Namen des zu löschenden Schlüssels zu löschen oder den Schlüssel mit der Option **-modify** zu ändern. Auf letztere muss der zu ändernde Schlüssel und der neue Schlüsselname folgen. Schließlich fügt die Option **-comment**, gefolgt von Text, dem FITS-Header einen Kommentar hinzu. Bitte beachten Sie, dass jeder Text, der Leerzeichen enthält, in doppelte Anführungszeichen eingeschlossen werden muss

visu

```
visu low high
```

Zeigt ein Bild mit **low** und **high** als untere und obere Schwelle an, nur GUI

wavelet

```
wavelet nbr_layers type
```

Computes the wavelet transform of the loaded image on (**nbr_layers**=1...n) layer(s) using linear (**type**=1) or bspline (**type**=2) version of the 'à trous' algorithm. The result is stored in a file as a structure containing the layers, ready for weighted reconstruction with WRECONS.

Siehe auch EXTRACT

Verweise: *wrecons*, *extract*

wiener

```
wiener [-loadpsf=] [-alpha=]
```

Stellt ein Bild mit Hilfe der Wiener Entfaltungsmethode wieder her.

Optional kann ein PSF (erzeugt mit MAKEPSF) mit dem Argument **-loadpsf=filename** geladen werden.

Der Parameter **-alpha=** gibt den Regularisierungsfaktor für Gaußsches Rauschen an

Verweise: *psf*, *makepsf*

wrecons

```
wrecons c1 c2 c3 ...
```

Rekonstruiert das aktuelle Bild aus den Layern, die zuvor mit waveelets berechnet und mit den Koeffizienten **c1**, **c2**, ..., **cn** entsprechend der Anzahl der für die Wavelet-Transformation verwendeten Layer gewichtet wurden, nach der Nutzung von WAVELET

Verweise: *wavelet*

KAPITEL 25

Sirilpy Python Module API 1.0.25 Reference

The sirilpy python module supports communication with a running Siril instance. It can request representations of the current loaded image plus its metadata, including details of detected stars, as well as the current loaded sequence and most frame metadata.

This documentation is autogenerated from version 1.0.25 of the python module code.

It can also run Siril commands using the `SirilInterface.cmd()` method, and the intent is to provide a capable interface for writing advanced scripts for Siril to a level not possible with the previous simple script files.

For example, scripts can now have PyQt6 or TKinter front ends using the PyQt6 or tkinter and ttkthemes modules, and they can utilise a lot of the ecosystem of Python modules including numpy, scipy, pillow and many more.

> **Bemerkung**
>
> There are some restrictions around modules that require installation of system binary packages for the modules to work.

In the initial module release, most methods relating to the image or sequence loaded in Siril are read-only. The intent is that the parameters of the loaded image or sequence can be obtained and used as inputs to scripts, for example for calculations or input to conditionals, but the existing mature Siril command set should in most cases be used to act on the loaded image. Thus header keywords can be set using `cmd("update_key", "key", "value")`, and most built-in image operations can be carried out using the appropriate Siril command. The main exception to the rule of python methods providing read-only access is the `set_image_pixeldata()` method, which allows for setting the pixel data in the loaded image from a numpy array. This means that new pixel processing algorithms can be added using python, initially getting the pixel data from the loaded image using `get_image_pixeldata()` and setting it on completion using `set_image_pixeldata()`. Similar functions are available for getting and setting pixel data from sequence frames.

25.1 Abhängigkeiten

Das sirilpy Modul benötigt folgendes:

- numpy >= 1.20.0
- packaging >= 21.0
- pywin32 >= 300 (only on Windows)

25.2 Siril Script Coding Guidance

- Unlike most python environments, Siril scripts run directly from Siril and the end user may not typically know how to access or how to use the python venv from outside Siril. This means that some tasks that are fairly trivial in typical python scenarios, such as installing packages, become harder. The user cannot be expected to access the command line and install packages themselves using `python3 -m pip install`.
  - The module therefore provides the `ensure_installed()` method. This uses pip to ensure that modules are installed so that they can be imported.
- Siril scripts should **always** provide the author's name, copyright details / license information and contact details such as a YouTube channel, website or forum where they can be contacted in regard to their script. This is advisory for independently distributed scripts and mandatory for any scripts that are submitted to the scripts repository. If you write a script, you are responsible for supporting it.
- As the repository grows and the API develops, not all the scripts published will necessarily always be compatible with all versions of Siril that will be in use:
  - If your script uses Siril commands, you should use the `requires` Siril command. This can be called directly in a Siril Script File, or it can be called from a Python script using `SirilInterface.cmd("requires", "min_version", {"max_version"})` (max_version is optional) but can be used to ensure scripts designed for older versions of Siril no longer show as applicable to newer versions if the command syntax has changed.
  - If your script uses any features of the Siril python module added since the initial release, you should call the `sirilpy.check_module_version()` method to ensure the installed version meets the requriements of your script. The versions at which features were added will be listed in the API documentation for all features added after the initial public release.

  - The code that populates the Siril view of the scripts repository will automatically filter out scripts whose Siril version or python module version requirements are not met.
  - If new classes or methods are added to the module after its initial public release the version at which they were introduced will be annotated in the docstring and the online documentation generated from it.
- Siril targets Linux, Windows and MacOS. Script writers are encouraged, where possible, to ensure that their scripts run correctly on all three OSes.

25.3 Sirilpy Connection

This submodule provides the main SirilInterface class used for communication between Siril and the python script. All its members are made available at the module root level, there is no need to import connection separately. Connection module for Siril, providing the ability to connect to a running Siril instance and communicate with it. Includes an extensive range of methods that can be used to get and set data from / to Siril.

class `sirilpy.connection.`**`SirilInterface`**

Bases: `object`

SirilInterface is the main class providing an interface to a running Siril instance and access to methods to interact with it through Siril's inbuilt command system and accessing image and sequence data.

`analyse_image_from_file`(*filepath*)

Request Siril to load an image from a file and analyse it. This method does not change the image currently loaded in Siril. Any image format supported by Siril is supported. An ImageAnalysis object is returned, containing parameters that may be used to assess the quality of an image for use in culling.

Parameter
`filepath` (`str`) -- String specifying the path to the image file to load.

Rückgabetyp
`ImageAnalysis`

Rückgabe
ImageAnalysis object containing quality metrics for the analysed image.

Verursacht

- **`FileNotFoundError`** -- if the specified file does not exist,
- *`DataError`* -- on receipt of incorrect data,
- *`SirilError`* -- if an error occurs.

`clear_image_bgsamples`()

Löscht alle Punkt des Hintergrundsamples vom Bild.

`clear_undo_history`()

Clears the image undo history, if there is undo history to be cleared

`cmd`(**args*)

Send a command to Siril to be executed. The range of available commands can be found by checking the online documentation. The command and its arguments are provided as a list of strings.

Parameter
`*args` (`str`) -- Variable number of string arguments to be combined into a command

Verursacht

- *`DataError`* -- If no response (or an incorrect response) was received,
- *`CommandError`* -- If the command returns an error status code,

- *SirilError* -- If any other error occurs during execution.

Beispiel

```
siril.cmd("ght", "-D=0.5", "-b=2.0")
```

command_error_message(*status_code*)

Provides a string describing the return status from a Siril command.

Parameter

status_code (*CommandStatus*) -- The status code returned by the Siril command handler, or by the CommandError exception.

Rückgabe

A string providing a description of the error code returned by a Siril command, for use in exception handling.

Rückgabetyp

str

confirm_messagebox(*title*, *message*, *confirm_label*)

Create a modal confirmation message dialog in Siril and wait for the response.

Parameter

- **title** (str) -- The title to display in the message box (up to 256 characters)
- **message** (str) -- The message to display in the message box (up to 1021 characters)
- **confirm_label** (str) -- The label to display in the message box confirmation button (OK, Yes, Confirm etc.) (Up to 24 characters)

Rückgabe

True if the message box confirmation button was clicked, False otherwise

Rückgabetyp

bool

Verursacht

- *DataError* -- if no response was received,
- *SirilError* -- if another error occurred.

connect()

Establish a connection to Siril based on the pipe or socket path.

Rückgabetyp

bool

Rückgabe

True on success

Verursacht

SirilConnectionError -- if a connection error occurred

create_new_seq(*seq_root*)

Creates a new .seq file with all images named seq_rootXXXXX.ext located in the current home folder. If a sequence with the same name is already loaded in Siril, it will not be recreated. This only works for FITS files, not FITSEQ nor SER. The newly created sequence is not loaded in Siril.

Parameter

seq_root (str) -- Der Stammname der zu erstellenden Sequenz.

Rückgabe
True if the sequence was successfully created, False otherwise.

Rückgabetyp
bool

Verursacht
SirilError -- if an error occurred.

`disconnect()`

Closes the established socket or pipe connection. Note there is not usually any need to close this unless for some reason you wish to close a connection and subsequently reopen another one. This method is automatically called at script termination using an `atexit` handler so there is no need to do so manually. Calling this method will reset the progress bar.

Verursacht
SirilConnectionError -- if the connection cannot be closed.

`error_messagebox(my_string, modal=False)`

Send an error message to Siril. The maximum message length is 1022 bytes: longer messages will be truncated (but this is more than enough for an error message box). Note that the error message box is not modal by default: this is intended for displaying an error message more prominently than using the Siril log prior to quitting the application.

Parameter

- **my_string** (`str`) -- The message to display in the error message box
- **modal** (`Optional[bool]`) -- Sets whether or not the message box should be modal and wait for completion or non-modal and allow the script to continue execution. Note that although a modal message box will block execution of the script, if a TKinter main loop is running events will continue to queue up, so if the message box is triggered by clicking a button then the user may click it while the message box is shown and trigger a second message box which will display immediately the first one is closed.

Rückgabe
True if the error was successfully displayed, False otherwise

Rückgabetyp
bool

Verursacht
SirilError -- if an error occurred.

`get_image(with_pixels=True, preview=False)`

Request a copy of the current image open in Siril. Requires a single image to be loaded.

Parameter

- **with_pixels** (`Optional[bool]`) -- optional bool specifying whether to get pixel data as a NumPy array, or only the image metadata. Defaults to True
- **preview** (`Optional[bool]`) -- optional bool specifying whether to get pixeldata as a preview (i.e. 8-bit autostretched data) or as real image data. Defaults to False (i.e. real image data)

Rückgabetyp
`Optional[FFit]`

Rückgabe
FFit object containing the image metadata and (optionally) pixel data, or None if an error occurred

Verursacht

- *NoImageError* -- wenn kein Bild in Siril geladen ist
- *SirilError* -- if a decoding error occurs

`get_image_bgsamples()`

Hintergrund von Siril anfordern.

Rückgabetyp
`Optional[List[BGSample]]`

Rückgabe
List of BGSamples background samples, with each set of coordinates expressed as a tuple[float, float], or None if no background samples have been set.

Verursacht

- *NoImageError* -- If no image is currently loaded,
- *DataError* -- on receipt of bad data,
- *SirilError* -- For other errors during data retrieval,

`get_image_filename()`

Request the filename of the loaded image from Siril. Requires a single image to be loaded.

Rückgabetyp
`Optional[str]`

Rückgabe
The filename as a string.

Verursacht

- *NoImageError* -- wenn kein Bild geladen ist,
- *SirilError* -- if a decoding error occurs.

`get_image_fits_header(return_as='str')`

Retrieve the full FITS header of the current image loaded in Siril. Requires a single image to be loaded.

Parameter
return_as -- Optional string specifying the format of the returned header. Can be 'str' for a string or 'dict' for a dictionary.

Rückgabe
The image FITS header as a string, or None if there is no header. dict: The image FITS header as a dictionary, or None if there is no header. None: If the header is empty or not available.

Rückgabetyp
str

Verursacht

- *NoImageError* -- If no image is currently loaded,
- *SirilError* -- For other errors during data retrieval,

`get_image_history()`

Retrieve history entries in the FITS header of the current loaded Siril image using shared memory. Requires a single image to be loaded.

Parameter
none.

Rückgabe

The HISTORY entries in the FITS header as a list of strings, or
None if there are no HISTORY keywords.

Rückgabetyp
list

Verursacht

- *NoImageError* -- If no image is currently loaded,
- *SirilError* -- For other errors during data retrieval.

`get_image_iccprofile()`

Retrieve the ICC profile of the current Siril image using shared memory. Requires a single image to be loaded.

Args: none.

Rückgabe
The image ICC profile as a byte array, or None if the current image has no ICC profile.

Rückgabetyp
bytes

Verursacht

- *NoImageError* -- If no image is currently loaded,
- *SirilError* -- If any other error occurs.

`get_image_keywords()`

Request FITS keywords data from Siril as a FKeywords object. Requires a single image to be loaded.

Rückgabetyp
`Optional[FKeywords]`

Rückgabe
FKeywords object containing the FITS keywords, or None if an error occurred

Verursacht
SirilError -- if a decoding error occurs.

`get_image_pixeldata(shape=None, preview=False, linked=False)`

Ruft die Pixeldaten aus dem aktuell in Siril geladenen Bild ab.

Parameter

- **shape** (`Optional[list[int]]`) -- Optional list of [x, y, w, h] specifying the region to retrieve. If provided, gets pixeldata for just that region. If None, gets pixeldata for the entire image.
- **preview** (`Optional[bool]`) -- optional bool specifying whether to get pixeldata as a preview (i.e. 8-bit autostretched data) or as real image data. Defaults to False (i.e. real image data)
- **linked** (`Optional[bool]`) -- optional bool specifying whether the autostretch preview should be linked or unlinked. If preview == False then this option is ignored.

Rückgabe
The image data as a numpy array

Rückgabetyp
numpy.ndarray

Verursacht

- *NoImageError* -- If no image is currently loaded,
- **ValueError** -- If an invalid shape is provided,
- *DataError* -- if the array cannot be reshaped to the correct dimensions,
- *SirilError* -- For other errors during pixel data retrieval,

get_image_shape()

Request the shape of the image from Siril.

Rückgabetyp
`Optional[Tuple[int, int, int]]`

Rückgabe
A tuple (channels, height, width) representing the shape of the image, or None if no image shape is available to return.

Raises: SirilError: if an error occurred.

get_image_stars(*channel=None*)

Request star model PSF data from Siril.

Parameter
channel (`Optional[int]`) -- Optional int specifying the channel to retrieve from. If provided 0 = Red / Mono, 1 = Green, 2 = Blue. If the channel is omitted the default behavior will be used: channel 0 for mono images, channel 1 (green) for color images. **channel requires sirilpy v1.0.8 or higher.**

Rückgabetyp
`List[PSFStar]`

Rückgabe
List of PSFStar objects containing the star data, or None if no stars can be found. If stars have already been detected using the *findstar* command then this list will be returned, otherwise automatic star detection will be attempted with the current star finder settings.

Verursacht

- *NoImageError* -- If no image is currently loaded,
- **ValueError** -- If an invalid channel is provided,
- *SirilError* -- For other errors during data retrieval,

get_image_stats(*channel*)

Request image statistics from Siril for a specific channel. Requires a single image to be loaded.

Parameter
channel (`int`) -- Integer specifying which channel to get statistics for (typically 0, 1, or 2)

Rückgabetyp
`Optional[ImageStats]`

Rückgabe
ImageStats object containing the statistics, or None if no stats are available for the selected channel

Verursacht

- *NoImageError* -- wenn kein Bild geladen ist,
- *SirilError* -- if an error occurs.

get_image_unknown_keys()

Retrieve the unknown key in a FITS header of the current loaded Siril image using shared memory. Requires a single image to be loaded.

Parameter
none.

Rückgabe
The unknown keys as a string, or None if there are no unknown keys.

Rückgabetyp
bytes

Verursacht

- *NoImageError* -- If no image is currently loaded,
- *SirilError* -- For other errors during data retrieval.

get_selection_star(*shape=None*, *channel=None*, *assume_centred=False*)

Retrieves a PSFStar star model from the current selection in Siril. Only a single PSFStar is returned: if there are more than one in the selection, the first one identified by Siril's internal star detection algorithm is returned. **Update**: from sirilpy 1.0.4 this method uses Siril's photometry functions to try to provide photometrically accurate values for PSFStar.mag, PSFStar.s_mag and PSFStar.SNR. If photometry succeeded and no saturated pixels were detected then PSFStar.phot_is_valid will be True, otherwise it will be False. :type shape: `Optional[list[int]]` :param shape: Optional list of [x, y, w, h] specifying the selection to

retrieve from. w x h must not exceed 300 px x 300 px. If provided, looks for a star in the specified selection If None, looks for a star in the selection already made in Siril, if one is made.

Parameter

- **channel** (`Optional[int]`) -- Optional int specifying the channel to retrieve from. If provided 0 = Red / Mono, 1 = Green, 2 = Blue. If the channel is omitted the current viewport will be used if in GUI mode, or if not in GUI mode the method will fall back to channel 0
- **assume_centred** (`bool`) -- Optional bool specifying whether to assume the star is already centred in the selection. Defaults to False.

Rückgabe

the PSFStar object representing the star model, or None if
in der Auswahl wurde kein Stern erkannt

Rückgabetyp
PSFStar

Verursacht

- **ValueError** -- If an invalid shape is provided,
- *NoImageError* -- If no image is loaded,
- *SirilConnectionError* -- If a communication error occurs,
- *SirilError* -- If any other error occurred during execution.

get_selection_stats(*shape=None*, *channel=None*)

Retrieves statistics for the current selection in Siril. Requires a single image or a sequence to be loaded.

Parameter

- **shape** (Optional[list[int]]) -- Optional list of [x, y, w, h] specifying the selection to retrieve from. If provided, looks for a star in the specified selection If None, looks for a star in the selection already made in Siril, if one is made.
- **channel** (Optional[int]) -- Optional int specifying the channel to retrieve from. If provided 0 = Red / Mono, 1 = Green, 2 = Blue. If the channel is omitted the current viewport will be used if in GUI mode, or if not in GUI mode the method will fall back to channel 0

Rückgabe
the ImageStats object representing the selection statistics.

Rückgabetyp
ImageStats

Verursacht

- *SirilError* -- Wenn während der Verarbeitung ein Fehler aufgetreten ist,
- **ValueError** -- If an invalid shape is provided.

get_seq()

Request metadata for the current sequence loaded in Siril.

Rückgabetyp
Optional[*Sequence*]

Rückgabe
Sequence object containing the current sequence metadata, or None if an error occurred

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *SirilError* -- if a decoding error occurs.

get_seq_distodata(*channel*)

Daten zur Sequenzverzerrung von Siril anfordern

channel: Integer specifying which channel to get registration data for (typically 0, 1, or 2)

Rückgabetyp
Optional[*DistoData*]

Rückgabe
DistoData object containing the channel distortion parameters, or None if an error occurred

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *SirilError* -- if a decoding error occurs.

get_seq_frame(*frame*, *with_pixels=True*, *preview=False*, *linked=False*)

Request sequence frame as a FFit from Siril. The keywords, statistics, header and other metadata are always returned: if an ICC profile is present, this will also be populated in the resulting FFit, and optionally the pixel data can also be returned.

Parameter

- **frame** (int) -- Integer specifying which frame in the sequence to retrieve data for (between 0 and Sequence.number - 1). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **with_pixels** (Optional[bool]) -- bool specifying whether or not to return the pixel data for the frame (default is True).

- **preview** (Optional[bool]) -- bool specifying whether or not to return the real pixel data or an autostretched uint8_t preview version. Only has an effect in conjunction with with_pixels = True

Rückgabetyp
Optional[*FFit*]

Rückgabe
FFit object containing the frame data

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *DataError* -- on receipt of incorrect data,
- *SirilError* -- if an error occurs.

get_seq_frame_filename(*frame*)

Request the filename of the specified frame of the loaded sequence from Siril. Requires a sequence to be loaded.

Parameter
frame (*int*) -- Specifies the frame index. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.

Rückgabetyp
Optional[str]

Rückgabe
The filename as a string.

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *SirilError* -- if a decoding error occurs.

get_seq_frame_header(*frame*, *return_as='str'*)

Rufen Sie den vollständigen FITS-Header eines Bildes aus der in Siril geladenen Sequenz ab.

Parameter

- **frame** (int) -- Integer specifying which frame in the sequence to retrieve data for (between 0 and Sequence.number - 1). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **return_as** -- Optional string specifying the format of the returned header. Can be 'str' for a string or 'dict' for a dictionary.

Rückgabe
The image FITS header as a string, or None if there is no header. dict: The image FITS header as a dictionary, or None if there is no header. None: If the header is empty or not available.

Rückgabetyp
str

Verursacht

- *NoSequenceError* -- If no sequence is currently loaded,
- *SirilError* -- For other errors during data retrieval,

get_seq_frame_pixeldata(*frame*, *shape=None*, *preview=False*, *linked=False*)

Ruft die Pixeldaten aus einem Bild der derzeit in Siril geladenen Sequenz ab.

Parameter

- **frame** (int) -- selects the frame to retrieve pixel data from. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **shape** (Optional[List[int]]) -- Optional list of [x, y, w, h] specifying the region to retrieve. If provided, gets pixeldata for just that region. If None, gets pixeldata for the entire image.
- **preview** (Optional[bool]) -- optional bool specifying whether to get pixeldata as a preview (i.e. 8-bit autostretched data) or as real image data. Defaults to False (i.e. real image data).
- **linked** (Optional[bool]) -- optional bool specifying whether the autostretched preview should be linked or unlinked. This option is ignored if preview is not True

Rückgabe

The image data as a numpy array

Rückgabetyp

numpy.ndarray

Verursacht

- **ValueError** -- If an invalid shape is provided,
- *DataError* -- if the array cannot be reshaped to the correct dimensions,
- *SirilError* -- For other errors during pixel data retrieval.

get_seq_imgdata(*frame*)

Request sequence frame metadata from Siril.

Parameter

frame (int) -- Integer specifying which frame in the sequence to get image metadata for (between 0 and Sequence.number). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.

Rückgabetyp

Optional[*ImgData*]

Rückgabe

ImgData object containing the frame metadata, or None if an error occurred

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *SirilError* -- if a decoding error occurs.

get_seq_regdata(*frame*, *channel*)

Request sequence frame registration data from Siril.

Parameter

- **frame** (int) -- Integer specifying which frame in the sequence to get registration data for (between 0 and Sequence.number). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **channel** (int) -- Integer specifying which channel to get registration data for (typically 0, 1, or 2)

Rückgabetyp
`Optional[RegData]`

Rückgabe
RegData object containing the registration data, or None if no registration data is available for the specified frame and channel

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *SirilError* -- if a decoding error occurs.

`get_seq_stats`(*frame*, *channel*)

Request sequence frame statistics from Siril.

Parameter

- **frame** (`int`) -- Integer specifying which frame in the sequence to get statistics data for (between 0 and Sequence.number). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **channel** (`int`) -- Integer specifying which channel to get statistics for (typically 0, 1, or 2)

Rückgabetyp
`Optional[ImageStats]`

Rückgabe
ImageStats object containing the statistics, or None if an error occurred

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- *SirilError* -- if a decoding error occurs.

`get_siril_active_vport`()

Request the active viewport from Siril.

Rückgabe

- sirilpy.SirilVport.RED / sirilpy.SirilVport.MONO
- sirilpy.SirilVport.GREEN,
- sirilpy.SirilVport.BLUE,
- sirilpy.SirilVport.RGB

Note that RED and MONO share the same IntEnum value, so there is no difference between a test for one and the other; the two enum labels are provided solely to aid code legibility.

Rückgabetyp
A SirilVport representing the active vport

Verursacht

- *DataError* -- if no response or an invalid response is received,
- *SirilError* -- if an error occurred.

`get_siril_config`(*group*, *key*)

Request a configuration value from Siril.

Parameter

- **group** (`str`) -- Konfiguration Gruppenname,

- **key** (`str`) -- Configuration key name within the group (Available values for group and key can be determined using the "get -A" command)

Rückgabetyp
`Union[bool, int, float, str, List[str], None]`

Rückgabe
The configuration value with appropriate Python type, or None if an error occurred.

Verursacht

- *DataError* -- if an unknown config type is encountered,
- *SirilError* -- if an error occurred getting the requested config value

`get_siril_configdir()`

Request the user config directory used by Siril.

Rückgabetyp
`str`

Rückgabe
The user config directory as a string.

Verursacht

- *DataError* -- if no response is received,
- *SirilError* -- for all other errors.

`get_siril_display_iccprofile()`

Retrieve the Siril display ICC profile.

Args: none.

Rückgabe
The display ICC profile as a byte array, or None if Siril is running in headless mode.

Rückgabetyp
bytes

Verursacht

- *NoImageError* -- If no image is currently loaded,
- *SirilError* -- If any other error occurs.

`get_siril_log()`

Retrieve the full Siril log as a text string.

Rückgabe
The text of the Siril log.

Rückgabetyp
str

Verursacht
SirilError -- For errors during data retrieval,

`get_siril_panzoom()`

Request the pan and zoom state from Siril.

Rückgabetyp
`Tuple[float, float, float]`

Rückgabe
A tuple (x offset, y offset, zoomlevel) representing the current image pan and zoom state.

Verursacht
SirilError -- if an error occurred.

`get_siril_selection()`

Request the image selection from Siril.

Rückgabetyp
`Optional[Tuple[int, int, int, int]]`

Rückgabe
A tuple (x, y, height, width) representing the current selection, or None if no selection is made.

Verursacht
SirilError -- if an error occurred.

`get_siril_slider_state(float_range=False)`

Request the display slider state from Siril.

Rückgabetyp
`Tuple[int, int, SlidersMode]`

Rückgabe
A tuple (min, max, slider mode) representing the current slider state.

Verursacht
SirilError -- if an error occurred.

`get_siril_stf()`

Request the Screen Transfer Function in use in Siril.

Rückgabetyp
STFType

Rückgabe
A STFType representing the current STF state.

Verursacht
SirilError -- if an error occurred.

`get_siril_stf_linked()`

Determine whether the AutoStretch STF is configured to be linked or not

Rückgabetyp
`bool`

Rückgabe
A bool representing the current STF channel-linked state.

Verursacht
SirilError -- if an error occurred.

`get_siril_systemdatadir()`

Fordern Sie das von Siril verwendete Systemdatenverzeichnis an.

Rückgabetyp
`Optional[str]`

Rückgabe
Systemdatenverzeichnis als Zeichenkette.

Verursacht

- *DataError* -- if no response is received,
- *SirilError* -- for all other errors.

get_siril_userdatadir()

Fordern Sie das von Siril verwendete Benutzerdatenverzeichnis an.

Rückgabetyp
str

Rückgabe
Das Benutzerdatenverzeichnis als Zeichenkette.

Verursacht

- *DataError* -- if no response is received,
- *SirilError* -- for all other errors.

get_siril_wd()

Fordern Sie das aktuelle Arbeitsverzeichnis von Siril an.

Rückgabetyp
str

Rückgabe
Das aktuelle Arbeitsverzeichnis als Zeichenkette

Verursacht

- *DataError* -- if no response was obtained,
- *SirilError* -- for all other errors.

image_lock()

A context manager that handles claiming and releasing the processing thread.

This method is designed to be used with a *with* statement to ensure that the thread is properly claimed before processing and released after processing, even if an exception occurs during processing. It is preferable to use this context manager rather than manually calling claim_thread() and release_thread() as the context manager will ensure correct cleanup if an exception occurs.

Note that the image_lock() context should only be entered when the script itself is operating on the Siril image data. If the script is calling a Siril command to alter the Siril image then the context **must not** be entered or the Siril command will be unable to acquire the processing thread and will fail.

Beispiel:

```
try:
    with siril.image_lock():
        # Get image data
        image_data = self.get_image_pixeldata()
        # Process image data
        processed_data = some_processing_function(image_data)
        # Set the processed image data
        siril.set_image_pixeldata(processed_data)
except ProcessingThreadBusyError:
    # Handle busy thread case
    pass
except ImageDialogOpenError:
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    # Handle open dialog case
    pass
```

Verursacht

- *ProcessingThreadBusyError* -- If the thread is already in use
- *ImageDialogOpenError* -- Wenn ein Bildbearbeitungsdialog geöffnet ist
- *SirilError* -- If the thread cannot be claimed or released for other reasons

info_messagebox(*my_string*, *modal=False*)

Send an information message to Siril. The maximum message length is 1022 bytes: longer messages will be truncated. This is intended for displaying informational messages more prominently than using the Siril log.

Parameter

- **my_string** (`str`) -- The message to display in the info message box
- **modal** (`Optional[bool]`) -- Sets whether or not the message box should be modal and wait for completion or non-modal and allow the script to continue execution. Note that although a modal message box will block execution of the script, if a TKinter main loop is running events will continue to queue up, so if the message box is triggered by clicking a button then the user may click it while the message box is shown and trigger a second message box which will display immediately the first one is closed.

Rückgabe
True if the info was successfully displayed, False otherwise

Rückgabetyp
bool

Verursacht
SirilError -- if an error occurred.

is_cli()

Check if the current instance is running in CLI mode. This method is useful to detect how the script was invoked and whether to show or not a GUI. This is False when the script is called by clicking in the Script menu, True otherwise.

Rückgabe
True if running in CLI mode, False otherwise.

Rückgabetyp
bool

is_image_loaded()

Check if a single image is loaded in Siril.

Rückgabe
True if a single image is loaded, False if a single image is not loaded

Rückgabetyp
bool

Verursacht

- *DataError* -- if no response is received,
- *SirilError* -- for all other errors.

is_sequence_loaded()

Check if a sequence is loaded in Siril.

Rückgabe
Wahr, wenn eine Sequenz geladen ist, Falsch wenn keine Sequenz geladen ist

Rückgabetyp
bool

Verursacht

- *DataError* -- if no response is received,
- *SirilError* -- for all other errors.

load_image_from_file(*filepath*, *with_pixels=True*, *preview=False*, *linked=False*)

Request Siril to load an image from a file and transfer it to sirilpy. This method does not change the image currently loaded in Siril. Any image format supported by Siril is supported. This may be used as an alternative to loading an image using astropy.io.fits, however perhaps the main benefit to using it is that it supports the preview option which can be used to obtain an 8-bit autostretched rendering of the image more quickly than is possible using astropy and applying an autostretch using numpy.

Parameter

- **filepath** (str) -- String specifying the path to the image file to load.
- **with_pixels** (Optional[bool]) -- bool specifying whether or not to return the pixel data for the image (default is True).
- **preview** (Optional[bool]) -- bool specifying whether or not to return the real pixel data or an autostretched uint8_t preview version. Only has an effect in conjunction with with_pixels = True
- **linked** (Optional[bool]) -- bool specifying whether the autostretch preview should be linked or not. Has no effect unless preview is True.

Rückgabetyp
Optional[*FFit*]

Rückgabe
FFit-Objekt, das die Bilddaten enthält

Verursacht

- **FileNotFoundError** -- if the specified file does not exist,
- *DataError* -- on receipt of incorrect data,
- *SirilError* -- if an error occurs.

log(*my_string*, *color=LogColor.DEFAULT*)

Send a log message to Siril. The maximum message length is 1022 bytes: longer messages will be truncated.

Parameter

- **my_string** (str) -- The message to log
- **color** (*LogColor*) -- Defines the text color, defaults to white. See the documentation
- **which** (*for LogColor for an explanation of which colors should be used for*)
- **purposes.**

Verursacht
SirilError -- wenn der Befehl fehlschlägt

Rückgabetyp
bool

open_dialog(*dialog*)

Opens a Siril GUI dialog. Introduced in sirilpy version 1.0.20.

Parameter
dialog (DialogID) -- Specifies the dialog to open.

Rückgabe
True if the dialog-opening action succeeded False if the dialog-opening action failed (e.g. because the criteria were not met)

Verursacht

- **TypeError** -- if the parameter is not a DialogID.
- *SirilError* -- if the method is called headless or an error occurs.

overlay_add_polygon(*polygon*)

Adds a user polygon to the Siril display overlay :type polygon: *Polygon* :param polygon: Polygon defining the polygon to be added

Rückgabe
the input updated with the ID assigned by Siril

Rückgabetyp
Polygon

Verursacht

- **TypeError** -- ungültige Daten geliefert,
- *SirilConnectionError* -- bei einem Verbindungsfehler,
- *DataError* -- on receipt of invalid data,
- *SirilError* -- bei jedem Fehler

overlay_clear_polygons()

Clears all user polygons from the Siril overlay

Rückgabe
True if the command succeeded, False otherwise

Rückgabetyp
bool

Verursacht

- *SirilError* -- if an error occurred
- *SharedMemoryError* -- if a shared memory error occurred

overlay_delete_polygon(*polygon_id*)

Deletes a single user polygon from the Siril overlay, specified by ID

Parameter
id -- int specifying the polygon ID to be deleted

Verursacht
SirilError -- on failure

overlay_draw_polygon(*color=16711744*, *fill=False*)

Enters a mode where the user can draw a Polygon in the Siril window by clicking the main mouse button and dragging. Releasing the mouse button finalises and closes the polygon.

Parameter

- **color** -- uint32 specifying packed RGBA values. Default: 0x00FF0040, 75% transparent green)
- **fill** -- bool specifying whether or not to fill the polygon (default: False)

overlay_get_polygon(*polygon_id*)

Gets a single user polygon from the Siril overlay, specified by ID

Parameter
id -- int specifying the polygon ID to be retrieved. The special ID -1 will

das zuletzt hinzugefügte Polygon abrufen.

Rückgabe
the specified Polygon if it exists, None otherwise

Rückgabetyp
Polygon

Verursacht
SirilError -- on failure

overlay_get_polygons_list()

Gets a List of all user polygons from the Siril overlay

Rückgabe
the list of Polygon if some exist, None otherwise

Rückgabetyp
List[*Polygon*]

Verursacht

- ***SirilError*** -- if an error occurred
- ***SharedMemoryError*** -- if a shared memory error occurred

pix2radec(*x*, *y*)

Converts a pair of pixel coordinates into RA and dec coordinates using the WCS of the image loaded in Siril. This requires that an image is loaded in Siril and that it has been platesolved (i.e. it has a WCS solution).

Parameter

- **x** (`float`) -- float: provides the x coordinate to be converted
- **y** (`float`) -- float: provides the y coordinate to be converted

Rückgabe
(RA, Dec) as a Tuple of two floats.

Rückgabetyp
Tuple[float, float]

Verursacht

- ***NoImageError*** -- If no image or sequence is loaded,
- **ValueError** -- If the image or loaded sequence frame is not plate solved,

- *SirilError* -- For errors during pix2radec execution.

radec2pix(*ra*, *dec*)

Converts a pair of RA,dec coordinates into image pixel coordinates using the WCS of the image loaded in Siril. This requires that an image is loaded in Siril and that it has been platesolved (i.e. it has a WCS solution).

Parameter

- **ra** (`float`) -- float: provides the RA coordinate to be converted
- **dec** (`float`) -- float: provides the dec coordinate to be converted

Rückgabe
[x, y] as a Tuple of two floats.

Rückgabetyp
Tuple[float, float]

Verursacht

- *NoImageError* -- If no image or sequence is loaded,
- **ValueError** -- If the image or loaded sequence frame is not plate solved,
- *SirilError* -- For errors during radec2pix execution.

redo()

Redoes the last undone operation, if there is an undo history and an undone operation available to be redone.

reset_progress()

Resets the Siril progress bar.

Parameter
none

Verursacht
SirilError -- Für eventuelle Fehler.

Rückgabetyp
`bool`

save_image_file(*data*, *header=None*, *filename=None*)

Save image pixeldata and metadata to a FITS file. This uses Siril to save the image and can therefore be used to avoid a script dependency on astropy if it is only required for saving an image. This allows images to be saved directly from python without having to use set_image_pixeldata() and set_image_metadata_from_header_string() and saving the Siril image. It support processing multiple images without affecting the image currently loaded in Siril.

Parameter

- **data** -- Either a numpy.ndarray containing the image data (must be 2D or 3D array with dtype float32 or uint16), OR a FFit object containing both data and header.
- **header** -- string containing the FITS header data. Required if data is a numpy array, ignored if data is a FFit object.
- **filename** -- string containing the path where the file should be saved. Required if data is a numpy array. If data is a FFit object and filename is None, will use fit.filename.

Rückgabe
True if successful, False otherwise

Rückgabetyp
bool

Verursacht

- **ValueError** -- if the input array or header is invalid,
- **TypeError** -- if invalid parameter types are provided,
- *SirilError* -- if there was an error in handling the command.

Beispiele

Using numpy array and header string siril.save_image_file(data_array, header_string, "output.fit")

Using FFit object siril.save_image_file(fit, filename="output.fit")

Using FFit object with its own filename siril.save_image_file(fit)

set_image_bgsamples(*points*, *show_samples=False*, *recalculate=True*)

Serialize a set of background sample points and send via shared memory. Points can be provided either as: - List of (x,y) Tuples: BGSamples are created with these positions and Siril will automatically compute the statistics. - List of BGSample objects: The complete sample data is sent to Siril. By default Siril will recalculate statistics for the sample points on receipt, but this can be overridden with the argument recalculate=False

> **Bemerkung**
>
> Siril 1.4.x does not require the processing thread to be claimed for this call, but from Siril 1.5.x onwards the request is refused unless the thread is claimed. It is good practice to call this method inside a `with image_lock():` block (or between `claim_thread()` and `release_thread()`) so that scripts remain forward compatible with 1.5.x and later.

Parameter

- **points** (`Union[List[Tuple[float, float]], List[BGSample]]`) -- List of sample points, either as (x,y) tuples or BGSample objects
- **show_samples** (`bool`) -- Whether to show the sample points in Siril
- **recalculate** -- Whether to recalculate the sample points once set. This only applies if the sample points are provided as a List of BGSamples, in which case it defaults to True. If the sample points are provided as a List of (x,y) Tuples then the parameter has no effect. Setting recalculate=False is usually a bad idea but the option is provided to support potential advanced uses where the values are adjusted in python code to manipulate the background fit.

Returns: True if the command succeeded, otherwise False

Verursacht

- *NoImageError* -- wenn kein Bild in Siril geladen ist,
- **ValueError** -- if samples do not have valid positions,
- *SirilError* -- if there was a Siril error in handling the command.

set_image_filename(*filename*)

Set the image filename in Siril. :type filename: `str` :param filename: the image filename to set :type filename: str

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the filename was set successfully

Rückgabetyp
bool

set_image_iccprofile(*iccprofile*)

Set the image ICC profile in Siril

> **Bemerkung**
>
> Siril 1.4.x does not require the processing thread to be claimed for this call, but from Siril 1.5.x onwards the request is refused unless the thread is claimed. It is good practice to call this method inside a `with image_lock():` block (or between `claim_thread()` and `release_thread()`) so that scripts remain forward compatible with 1.5.x and later.

Parameter

- **iccprofile** (*bytes*) -- The ICC profile to send to Siril. This will
- **profile** (*replace an existing ICC*)
- **None** (*if one is set. If*)

:param : :param any existing image ICC profile will be removed.:

Returns: True if the command succeeded, otherwise False

Verursacht

- *NoImageError* -- wenn kein Bild in Siril geladen ist,
- **ValueError** -- if iccprofile is not valid type,
- *SirilError* -- if there was a Siril error in handling the command.

set_image_metadata_from_header_string(*header*)

Send image metadata to Siril from a FITS header. The header can be obtained from a sirilpy FFit.header or alternatively from a FITS file obened from disk using astropy.fits.

Beispiel:

```
hdul = fits.open('your_fits_file.fits')
# Get the header from the primary HDU (or any other HDU you want)
header = hdul[0].header
# Convert the header to string
header_string = header.tostring(sep='\\n')
# Send the metadata to Siril
siril.set_image_metadata_from_header_string(header_string)
```

> **Bemerkung**
>
> Siril 1.4.x does not require the processing thread to be claimed for this call, but from Siril 1.5.x onwards the request is refused unless the thread is claimed. It is good practice to call this method inside a `with image_lock():` block (or between `claim_thread()` and `release_thread()`) so that scripts remain forward compatible with 1.5.x and later.

Parameter
header (`str`) -- Zeichenkette die die FITS Header Daten enthält

Rückgabe
True if successful, False otherwise

Rückgabetyp
bool

Verursacht

- **TypeError** -- Ungültiger Parameter angegeben,
- *NoImageError* -- wenn kein Bild in Siril geladen ist,
- *SirilError* -- if an error occurs.

set_image_pixeldata(*image_data*)

Send image data to Siril using shared memory.

> **Bemerkung**
>
> This method modifies the loaded image and requires the processing thread to be claimed: call it inside a `with image_lock():` block (or between `claim_thread()` and `release_thread()`), otherwise Siril will refuse the request with an error.

Parameter
image_data (`ndarray`) -- numpy.ndarray containing the image data. Must be 2D (single channel) or 3D (multi-channel) array with dtype either np.float32 or np.uint16.

Verursacht

- *NoImageError* -- wenn kein Bild in Siril geladen ist,
- **ValueError** -- if the input array is invalid,
- *SirilError* -- if there was an error in handling the command.

Rückgabetyp
`bool`

set_seq_frame_incl(*index*, *incl*)

Set whether given frame(s) are included in the currently loaded sequence in Siril. This method is intended for use in creating custom sequence filters.

Parameter

- **index** (`Union[int, List[int]]`) -- integer or list of integers specifying which frame(s) to set the inclusion status for. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame number 1. Passing a list is available since sirilpy 1.0.17

- **incl** (bool) -- bool specifying whether the frame(s) are included or not.

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- **TypeError** -- if index is not an int or list of ints,
- *SirilError* -- on failure.

set_seq_frame_pixeldata(*index*, *image_data*, *prefix*)

Send sequence frame image data to Siril using shared memory. Note that this method only works with sequences of FITS images: it does **not** work with FITSEQ, SER or AVI single-file sequences. The image_lock() context manager is not required in order to use this method.

Parameter

- **index** (int) -- integer specifying which frame to set the pixeldata for. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **image_data** (ndarray) -- numpy.ndarray containing the image data. Must be 2D (single channel) or 3D (multi-channel) array with dtype either np.float32 or np.uint16.
- **prefix** (str) -- String prefix to use when saving the file to make a new sequence. Note that saving sequence frames with a new prefix does not by itself create a new sequence: once all the frames have been saved with the new sequence prefix, `sirilpy.SirilInterface.create_new_seq()` must be called to create the actual sequence file. Note that while it is permitted to pass prefix=None, this will overwrite the existing sequence and is not typically what is wanted, therefore the parameter is not optional and must be passed explicitly.

Verursacht

- *NoSequenceError* -- if no sequence is loaded in Siril,
- **ValueError** -- if the input array is invalid,
- *SirilError* -- if there was an error in handling the command.

Rückgabetyp
bool

set_siril_pan(*xoff*, *yoff*)

Set the display offset (pan) in Siril to (xoff, yoff). :type xoff: `float` :param xoff: x display offset :type xoff: float :type yoff: `float` :param yoff: y display offset :type yoff: float

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the display offset was set successfully

Rückgabetyp
bool

set_siril_selection(*x=None*, *y=None*, *w=None*, *h=None*, *selection=None*)

Set the image selection in Siril using the provided coordinates and dimensions.

Parameter

- **x** (Optional[int]) -- X-coordinate of the selection's top-left corner (must be provided with y, w, h)

- **y** (Optional[int]) -- Y-coordinate of the selection's top-left corner (must be provided with x, w, h)
- **w** (Optional[int]) -- Width of the selection (must be provided with x, y, h)
- **h** (Optional[int]) -- Height of the selection (must be provided with x, y, w)
- **selection** (Optional[Tuple[int, int, int, int]]) -- A tuple of (x, y, w, h) as returned by get_siril_selection()

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the selection was set successfully

Rückgabetyp
bool

set_siril_slider_lohi(*lo=None*, *hi=None*)

Set the slider values in Siril using the provided lo and hi values. If the sliders mode is not already set to USER, it is set to that mode as setting slider values is only relevant in that mode. :type lo: Union[float, int] :param lo: Low value for the slider (float [0,1] or uint16) :type hi: Union[float, int] :param hi: High value for the slider (float [0,1] or uint16)

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the slider state was set successfully

Rückgabetyp
bool

set_siril_slider_mode(*mode*)

Set the slider state in Siril using the provided lo, hi and mode values. :type mode: *SlidersMode* :param mode: SlidersMode enum value

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the slider state was set successfully

Rückgabetyp
bool

set_siril_stf(*mode*)

Set the screen transfer function in Siril using the provided enum value. :type mode: *SlidersMode* :param mode: STFType enum value

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the slider state was set successfully

Rückgabetyp
bool

`set_siril_stf_linked`(*state*)

Set the screen transfer function in Siril using the provided enum value. :type state: `bool` :param state: bool

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the slider state was set successfully

Rückgabetyp
bool

`set_siril_zoom`(*zoom*)

Set the display zoom. Passing any negative value will set 'zoom to fit'. :type zoom: `float` :param zoom: zoom level :type zoom: float

Verursacht

- *SirilError* -- if an error occurred.
- **ValueError** -- if parameters are not properly provided.

Rückgabe
True if the display offset was set successfully

Rückgabetyp
bool

`undo`()

Undoes the last operation, if there is an undo history available.

`undo_save_state`(*my_string*)

Saves an undo state. The maximum message length is 70 bytes: longer messages will be truncated. Requires a single image to be loaded.

Parameter
my_string (`str`) -- The message to log in FITS HISTORY

Rückgabe
True if the message was successfully logged, False if a logging error occurred or None if the script is headless (undo does not happen when headless)

Rückgabetyp
Optional[bool]

Verursacht
SirilError -- if an error occurred.

`update_progress`(*message*, *progress*)

Send a progress update to Siril with a message and completion percentage.

Parameter

- **message** (`str`) -- Status message to display,

- **progress** (`float`) -- Progress value in the range 0.0 to 1.0. The following special values can be used: -1.0 will pulsate the progress bar, and -2.0 will update the progress bar text but will not update the progress shown in the bar.

Verursacht

- **ValueError** -- If the progress argument is out of range,
- *SirilError* -- For any other errors.

Rückgabetyp
`bool`

warning_messagebox(*my_string*, *modal=False*)

Send a warning message to Siril. The maximum message length is 1022 bytes: longer messages will be truncated. This is intended for displaying warning messages more prominently than using the Siril log.

Parameter

- **my_string** (`str`) -- The message to display in the warning message box
- **modal** (`Optional[bool]`) -- Sets whether or not the message box should be modal and wait for completion or non-modal and allow the script to continue execution. Note that although a modal message box will block execution of the script, if a TKinter main loop is running events will continue to queue up, so if the message box is triggered by clicking a button then the user may click it while the message box is shown and trigger a second message box which will display immediately the first one is closed.

Rückgabe
True if the warning was successfully displayed, False otherwise

Rückgabetyp
bool

Verursacht
SirilError -- if an error occurred.

xy_plot(*plot_data*, *display=True*, *save=False*)

Serialize plot data and send via shared memory. See the sirilpy.plot submodule documentation for how to configure a PlotData object for use with SirilInterface.xy_plot()

Parameter

- **plot_metadata** -- PlotMetadata object containing plot configuration
- **display** -- bool indicating whether to display the plot on screen (defaults to True)
- **save** -- bool indicating whether to save to the file specified in PlotData.savename (defaults to False)

Verursacht

- *DataError* -- if invalid xy_plot data is received via shared memory,
- *SirilError* -- If an error occurs.

25.4 Sirilpy Datenmodelle

This submodule provides dataclasses to represent the main Siril data structures. Most dataclasses have corresponding deserialization methods that are used by the SirilInterface methods. All its members are made available at the module root level, there is no need to import models separately.

class sirilpy.models.**BGSample**(*x=None*, *y=None*, *position=None*, *size=25*, ***kwargs*)

Bases: `object`

Python equivalent of the Siril background_sample struct. Used to hold background sample data obtained from Siril, or to generate or modify background sample data to set in Siril. A BGSample can be constructed as: - s1 = BGSample(x=1.0, y=2.0) - s2 = BGSample(position=(1.0, 2.0)) - s3 = BGSample(x=1.0, y=2.0, mean=0.5, size=31)

classmethod deserialize(*data*)

Deserialize a portion of a buffer into a BGSample object

Parameter
data (*bytes*) -- The full binary buffer containing BGSample data

Rückgabe
A BGSample object

Rückgabetyp
BGSample

Verursacht

- **ValueError** -- If the buffer slice size does not match the expected size.
- **struct.error** -- If there is an error unpacking the binary data.

max: float = 0.0

mean: float = 0.0

median: Tuple[float, float, float] = (0.0, 0.0, 0.0)

Median values for R, G and B channels. For mono images only median[0] is used.

min: float = 0.0

position: Tuple[float, float] | None = None

Position in (x, y) image coordinates

size: int = 25

The default size matches the size of Siril bg samples.

valid: bool = True

Samples default to being valid

class sirilpy.models.**DistoData**(*index=DistoType.DISTO_UNDEF*, *filename=''*, *velocity=(0, 0)*)

Bases: `object`

Python equivalent of Siril disto_params structure

filename: str = ''

filename if DISTO_FILE or DISTO_MASTER (and optional for DISTO_FILE_COMET)

index: *DistoType* = 0

Specifies the distrosion type

velocity: Tuple[float, float] = (0, 0)

shift velocity if DISTO_FILE_COMET

class sirilpy.models.**FFit**(*bitpix=None*, *orig_bitpix=None*, *naxis=0*, *_naxes=(0, 0, 0)*, *keywords=<factory>*, *checksum=False*, *header=None*, *unknown_keys=None*, *stats=<factory>*, *mini=0.0*, *maxi=0.0*, *neg_ratio=0.0*, *_data=None*, *top_down=False*, *_focalkey=False*, *_pixelkey=False*, *history=<factory>*, *color_managed=False*, *_icc_profile=None*)

Bases: `object`

Python equivalent of Siril ffit (FITS) structure, holding image pixel data and metadata.

allocate_data()

Allocate memory for image data with appropriate type. self.width, self.height, self.naxis, self.naxes and self.dtype must be set before calling this method.

Verursacht
ValueError -- if self.bitpix is not set to BitpixType.USHORT_IMG or BitpixType.FLOAT_IMG

ensure_data_type(*target_type=None*)

Ensure data is in the correct type with proper scaling

Parameter
target_type -- Optional np.dtype to convert to. If None, uses self.type

Verursacht
ValueError -- if the conversion is between data types that are not internally used by Siril for calculation

estimate_noise(*array*, *nullcheck=True*, *nullvalue=0.0*)

Estimate the background noise in the input image using the sigma of first-order differences.

noise = 1.0 / sqrt(2) * RMS of (flux[i] - flux[i-1])

Parameter

- **array** (*np.ndarray*) -- 2D array of image pixels (np.uint16 or np.float32).
- **nullcheck** (*bool*) -- If True, check for null values.
- **nullvalue** (`Optional[float]`) -- The value of null pixels (only used if nullcheck is True).

Rückgabe
Estimated noise value.

Rückgabetyp
float

Verursacht
ValueError -- if the array is the wrong shape

get_channel(*channel*)

Get a specific channel of the pixel data. Note that this does not pull pixel data directly from the image loaded in Siril: that must previously have been obtained using get_image_pixeldata() or get_image()

Rückgabetyp
`ndarray`

update_stats()

Update image statistics for all channels. Note that this only updates the statistics based on the NumPy array representing pixel data in the python FFit object, it does not update the statistics of the image in Siril.

bitpix: *BitpixType* | None = None

FITS-Header-Spezifikation des Bilddatentyps.

property channels: int

Bildkanäle

checksum: bool = False

Whether Siril will write FITS data checksums for this file.

color_managed: bool = False

Specifies whether the image is color managed or not.

property data: ndarray | None

The pixel data of the current image loaded in Siril, stored as a NumPy array

header: str | None = None

The FITS header as a string.

property height: int

Bildhöhe

history: list[str]

Contains a list of strings holding the HISTORY entries for this image.

property icc_profile: bytes | None

The ICC profile as raw bytes data. This may be converted for use by modules such as pillow which can handle ICC profiles.

keywords: *FKeywords*

A FKeywords object containing FITS header keywords.

maxi: float = 0.0

The maximum value across all image channels.

mini: float = 0.0

The minimum value across all image channels.

property naxes: Tuple[int, int, int]

The naxes tuple holds the image dimensions as width x height x channels. Note that the axis ordering differs between Siril representation as held in naxes and numpy representation as held in _data.shape (which is channels x height x width)

naxis: int = 0

The number of axes (2 for a mono image, 3 for a RGB image). Corresponds to the FITS kwyword NAXIS.

neg_ratio: float32 = 0.0

The ratio of negative pixels to the total pixels.

orig_bitpix: *BitpixType* | None = None

FITS header specification of the original image data type.

stats: List[*ImageStats* | None]

A list of ImageStats objects, one for each channel.

top_down: bool = False

Specifies the ROWORDER for this image. The FITS specification directs that FITS should be stored bottom-up, but many CMOS sensors are natively TOP_DOWN and capture software tends to save FITS images captured by these sensors as TOP_DOWN.

unknown_keys: str | None = None

All unknown FITS header keys as a string. This gives access to header cards that Siril does not use internally.

property width: int

Bildbreite

class sirilpy.models.**FKeywords**(*bscale=1.0*, *bzero=0.0*, *lo=0*, *hi=0*, *flo=0.0*, *fhi=0.0*, *program=''*, *filename=''*, *row_order=''*, *filter=''*, *image_type=''*, *object=''*, *instrume=''*, *telescop=''*, *observer=''*, *bayer_pattern=''*, *sitelat_str=''*, *sitelong_str=''*, *focname=''*, *date=None*, *date_obs=None*, *data_max=0.0*, *data_min=0.0*, *pixel_size_x=0.0*, *pixel_size_y=0.0*, *binning_x=1*, *binning_y=1*, *expstart=0.0*, *expend=0.0*, *bayer_xoffset=0*, *bayer_yoffset=0*, *airmass=1.0*, *focal_length=0.0*, *flength=0.0*, *iso_speed=0.0*, *exposure=0.0*, *aperture=0.0*, *ccd_temp=0.0*, *set_temp=0.0*, *livetime=0.0*, *stackcnt=0*, *cvf=0.0*, *gain=0*, *offset=0*, *focuspos=0*, *focussz=0*, *foctemp=0.0*, *centalt=0.0*, *centaz=0.0*, *sitelat=0.0*, *sitelong=0.0*, *siteelev=0.0*, *objctra=None*, *objctdec=None*, *ra=None*, *dec=None*, *pltsolvd=False*, *pltsolvd_comment=None*)

Bases: *object*

Python equivalent of Siril fkeywords structure. Contains the FITS header keyword values converted to suitable data types.

classmethod deserialize(*data*)

Deserialize binary keyword-block (the block produced by keywords_to_py()) into an FKeywords object. Raises ValueError on size mismatch and SirilError for other unpacking issues.

Rückgabetyp
FKeywords

FLEN_VALUE: ClassVar[int] = 71

KEYWORDS_FORMAT: ClassVar[str] = '!71s71s71s71s71s71s71s71s71s71s71s71s71s71s71sddQQddddddQQddddddddqqddddddddddQdqqqqdqqdd?'

KEYWORDS_SIZE: ClassVar[int] = 1465

airmass: float = 1.0

Luftmasse (Airmass) in der Bildmitte (Gueymard 1993)

aperture: float = 0.0

Aperture value as a float

bayer_pattern: str = ''

Bayermuster

bayer_xoffset: int = 0

X-Versatz des Bayer-Musters

bayer_yoffset: int = 0

Y-Versatz des Bayer-Musters

binning_x: int = 1

XBINNING FITS header card as an int

binning_y: int = 1

YBINNING FITS header card as an int

bscale: float = 1.0

Datenbereich Offset zu dem von Vorzeichenlosen Shorts verschieben

bzero: float = 0.0

Standard-Skalierungsfaktor

ccd_temp: float = 0.0

CCD Temperatur als Flieskommazeichen

centalt: float = 0.0

[deg] Höhe/Alt des Teleskops

centaz: float = 0.0

[deg] Azimuth des Teleskops

cvf: float = 0.0

Konvertierungsfaktor (e-/ADU)

data_max: float = 0.0

used to check if 32b float is in the [0, 1] range

data_min: float = 0.0

used to check if 32b float is in the [0, 1] range

date: datetime | None = None

Datum zu dem das FITS erstellt wurde als UTC-Datum

date_obs: datetime | None = None

ss Beobachtungsbeginn, UT

Type
YYYY-MM-DDThh

Type
mm

dec: float | None = None

Dec as float, if available

expend: float = 0.0

Belichtungsende als julianisches Datum

exposure: float = 0.0

Exposure time as a float (s)

expstart: float = 0.0

Belichtungsbeginn als julianisches Datum

fhi: float32 = 0.0

MIPS-Hi key in FITS file, "Upper visualization cutoff (float)"

filename: str = ''

Originaler Dateiname

filter: str = ''

Name des genutzten Filters

flength: float = 0.0

[mm] Brennweite

flo: float32 = 0.0

MIPS-LO key in FITS file, "Lower visualization cutoff (float)"

focal_length: float = 0.0

[mm] Brennweite

focname: str = ''

Name des Fokusers

foctemp: float = 0.0

Temperatur des Fokussierers

focuspos: int = 0

Position des Fokussierers

focussz: int = 0

[um] Schrittweite des Fokusers

gain: int = 0

Gain factor read in camera

hi: int = 0

MIPS-HI-Schlüssel in FITS-Datei, „Obere Visualisierungsgrenze“

image_type: str = ''

Bildtyp

instrume: str = ''

Name des Instruments

iso_speed: float = 0.0

ISO speed value as a float

livetime: float = 0.0

Summe der Belichtungszeiten [s]

lo: int = 0

MIPS-LO-Schlüssel in FITS-Datei, „Unterer Visualisierungsgrenzwert“

objctdec: str | None = None

object Dec as string, if available

objctra: str | None = None

object RA as string, if available

object: str = ''

Name des interessierenden Objekts

observer: str = ''

Name des Beobachters

offset: int = 0

Offset value read in camera

pixel_size_x: float = 0.0

XPIXSZ FITS header card as a float

pixel_size_y: float = 0.0

YPIXSZ FITS header card as a float

pltsolvd: bool = False

whether plate solved or not

pltsolvd_comment: str | None = None

plate solution comment

program: str = ''

Software, die diese HDU erstellt hat

ra: float | None = None

RA as float, if available

row_order: str = ''

Reihenfolge der Zeilen in einem Bildarray

set_temp: float = 0.0

CCD-Temperatur als Kommazahl einstellen

siteelev: float = 0.0

[m] Höhe des Beobachtungsortes

sitelat: float = 0.0

sitelat_str: str = ''

sitelong: float = 0.0

[Grad] Längengrad des Beobachtungsortes

sitelong_str: str = ''

stackcnt: int = 0

Anzahl gestackter Bilder

telescop: str = ''

Name des für die Aufnahme genutzten Teleskops

class sirilpy.models.**FPoint**(*x*, *y*)

Bases: object

Represents a 2D point with float x and y coordinate values in the Siril image.

x: float

x Koordinate

y: float

y Koordinate

class sirilpy.models.**Homography**(*h00=0.0*, *h01=0.0*, *h02=0.0*, *h10=0.0*, *h11=0.0*, *h12=0.0*, *h20=0.0*, *h21=0.0*, *h22=0.0*, *pair_matched=0*, *Inliers=0*)

Bases: object

Python equivalent of the Siril Homography structure. Contains coefficients for the Homography matrix that maps a sequence frame onto the reference frame.

Inliers: int = 0

number of inliers kept after RANSAC step

h00: float = 0.0

Homographie matrix H00

h01: float = 0.0

Homographie matrix H01

h02: float = 0.0

Homographie matrix H02

h10: float = 0.0

Homographie matrix H10

h11: float = 0.0

Homographie matrix H11

h12: float = 0.0

Homographie matrix H12

h20: float = 0.0

Homographie matrix H20

h21: float = 0.0

Homographie matrix H21

h22: float = 0.0

Homographie matrix H22

pair_matched: int = 0

number of pairs matched

class sirilpy.models.**ImageAnalysis**(*bgnoise=0.0*, *fwhm=0.0*, *wfwhm=0.0*, *nbstars=0*, *roundness=0.0*, *imagetype=0*, *timestamp=0*, *channels=0*, *height=0*, *width=0*, *filter=''*)

Bases: `object`

Structure to hold image analysis data, used for culling

classmethod deserialize(*data*)

Unpack a network-safe binary struct into an ImageAnalysis instance.

Rückgabetyp
ImageAnalysis

serialize()

Pack the dataclass into a network-safe binary struct.

Rückgabetyp
`bytes`

FLEN: ClassVar[int] = 71

bgnoise: float = 0.0

RMS Hintergrundrauschen

channels: int = 0

Anzahl der Kanäle im Bild

filter: str = ''

filter name (fixed-length string from C, max 70 chars)

fwhm: float = 0.0

mittlere FWHM

height: int = 0

Bildhöhe

imagetype: *ImageType* = 0

Image type enum

nbstars: int = 0

Anzahl der erkannten Sterne

roundness: float = 0.0

Mean star roundness

timestamp: int = 0

00 UTC)

Type
UNIX timestamp (64-bit seconds since 1970/1/1 00

wfwhm: float = 0.0

Mean weighted fwhm

width: int = 0

Bildbreite

class sirilpy.models.**ImageStats**(*total=0*, *ngoodpix=0*, *mean=0.0*, *median=0.0*, *sigma=0.0*, *avgDev=0.0*, *mad=0.0*, *sqrtbwmv=0.0*, *location=0.0*, *scale=0.0*, *min=0.0*, *max=0.0*, *normValue=0.0*, *bgnoise=0.0*)

Bases: `object`

Python equivalent of Siril imstats structure. Contains statistics for a particular channel of a Siril image.

classmethod deserialize(*data*)

Deserialize binary data into an ImageStats object.

Parameter
data (`bytes`) -- (bytes) Binary data to unpack

Rückgabetyp
ImageStats

Returns: ImageStats object

Raises: SirilError: If received data size is incorrect
struct.error: If unpacking fails

avgDev: float = 0.0

Durchschnittliche Abweichung der Pixel

bgnoise: float = 0.0

RMS Hintergrundrauschen

location: float = 0.0

Position der Pixelwerte

mad: float = 0.0

mittlere durchschnittliche Abweichung der Pixelwerte

max: float = 0.0

Maximale Pixelwerte

mean: float = 0.0
Mittelwert der Pixel

median: float = 0.0
median value of pixels

min: float = 0.0
mInimaler Pixelwert

ngoodpix: int = 0
number of non-zero pixels

normValue: float = 0.0
norm value of the pixels

scale: float = 0.0
Skalierungswert der Pixel

sigma: float = 0.0
Standardabweichung der Pixelwerte

sqrtbwmv: float = 0.0
Quadratwurzel der Biweight-Mittelsvarianz der Pixelwerte

total: int = 0
total number of pixels

class sirilpy.models.**ImgData**(*filenum=0*, *incl=False*, *date_obs=None*, *airmass=0.0*, *rx=0*, *ry=0*)

Bases: `object`

Python equivalent of Siril imgdata structure

classmethod deserialize(*response*)

Deserialize binary response into an ImgData object.

Parameter
response (*bytes*) -- Binary data to unpack.

Rückgabe
An ImgData object with deserialized data.

Rückgabetyp
ImgData

Verursacht

- **ValueError** -- If received data size is incorrect.
- **struct.error** -- If unpacking fails.

airmass: float = 0.0
Luftmasse des Bildes

date_obs: datetime | None = None
Datum der Beobachtung

filenum: int = 0
real file index in the sequence

incl: bool = False
in der Sequenz ausgewählt

rx: int = 0

width

ry: int = 0

height

class sirilpy.models.**PSFStar**(*star_name=None*, *B=0.0*, *A=0.0*, *x0=0.0*, *y0=0.0*, *sx=0.0*, *sy=0.0*, *fwhmx=0.0*, *fwhmy=0.0*, *fwhmx_arcsec=0.0*, *fwhmy_arcsec=0.0*, *angle=0.0*, *rmse=0.0*, *sat=0.0*, *R=0*, *has_saturated=False*, *beta=0.0*, *profile=StarProfile.GAUSSIAN*, *xpos=0.0*, *ypos=0.0*, *mag=0.0*, *Bmag=0.0*, *s_mag=999.99*, *s_Bmag=999.99*, *SNR=0.0*, *phot_is_valid=False*, *BV=0.0*, *B_err=0.0*, *A_err=0.0*, *x_err=0.0*, *y_err=0.0*, *sx_err=0.0*, *sy_err=0.0*, *ang_err=0.0*, *beta_err=0.0*, *layer=0*, *units=None*, *ra=0.0*, *dec=0.0*)

Bases: `object`

Python equivalent of the Siril fwhm_struct structure. Contains data on a modelled fit to a star identified in the image.

classmethod deserialize(*data*)

Deserialize a portion of a buffer into a PSFStar object.

Parameter
data (`bytes`) -- (bytes) The full binary buffer containing PSFStar data.

Rückgabetyp
PSFStar

Rückgabe
PSFStar Objekt

Verursacht

- **ValueError** -- If the buffer slice size does not match the expected size.
- **struct.error** -- If there is an error unpacking the binary data.

A: float = 0.0

Amplitude

A_err: float = 0.0

error in A

B: float = 0.0

Durchschnittlicher Wert des Himmelshintergrundes

BV: float = 0.0

wird nur zur Übertragung von Daten bei der photometrischen Farbkalibrierung verwendet

B_err: float = 0.0

error in B

Bmag: float = 0.0

B magnitude

R: int = 0

Optimized box size to enclose sufficient pixels in the background

SNR: float = 0.0

SNR of the star

ang_err: float = 0.0

error in angle

angle: float = 0.0

angle of the x and yaxes with respect to the image x and y axes

beta: float = 0.0

Moffat-Gleichung Beta-Parameter

beta_err: float = 0.0

error in beta

dec: float = 0.0

Deklination

fwhmx: float = 0.0

FWHM in x axis in pixels

fwhmx_arcsec: float = 0.0

FWHM in x axis in arc seconds

fwhmy: float = 0.0

FWHM in y axis in pixels

fwhmy_arcsec: float = 0.0

FWHM in y axis in arc seconds

has_saturated: bool = False

Shows whether the star is saturated or not

layer: int = 0

image channel on which the star modelling was carried out

mag: float = 0.0

(V) magnitude, approximate or accurate

phot_is_valid: bool = False

True if mag, Bmag, s_mag, s_Bmag and SNR are photometrically accurate with no saturated pixels. Introduced in sirilpy 1.0.4.

profile: *StarProfile* = 0

ra: float = 0.0

Rektaszension

rmse: float = 0.0

RMSE der Minimierung

s_Bmag: float = 999.99

Fehler der B Helligkeit

s_mag: float = 999.99

error on the (V) magnitude

sat: float = 0.0

Level above which pixels have satured

star_name: str | None = None

sx: float = 0.0

Size of the fitted function on the x axis in PSF coordinates

sx_err: float = 0.0

error in sx

sy: float = 0.0

Size of the fitted function on the y axis in PSF coordinates

sy_err: float = 0.0

error in sy

units: str | None = None

Units

x0: float = 0.0

x-Koordinate des Peaks

x_err: float = 0.0

error in x

xpos: float = 0.0

X-Position des Sternes im Bild

y0: float = 0.0

Y-Koordinate des Peaks

y_err: float = 0.0

error in y

ypos: float = 0.0

Y-Position des Sterns im Bild

class sirilpy.models.**Polygon**(*points*, *polygon_id=0*, *color=4294967295*, *fill=False*, *legend=None*)

Bases: `object`

Represents a user-defined polygon. These can be filled or outline-only, and can have any color and transparency (alpha) value. They can also have an optional label which is displayed centred on the polygon.

Note that Polygons should be considered transitory if used with the overlay - they can be used to display information to the user but they may be cleared at any time if the user toggles the overlay button in the main Siril interface to clear the overlay.

classmethod deserialize_polygon(*data*)

Creates a Polygon object by deserializing a byte array.

Rückgabe

A Polygon object and any remaining bytes in the byte
array. (The remaining bytes are for use in deserialize_polygon_list and can be safely ignored if deserializing a single polygon.)

Rückgabetyp
Tuple

Verursacht
`ValueError` -- If there is insufficient data to deserialize.

classmethod deserialize_polygon_list(*data*)

Creates a List of Polygon objects by deserializing a byte array.

Rückgabe
A List of Polygon objects.

Rückgabetyp
List

Verursacht
ValueError -- If there is invalid data to deserialize.

classmethod from_rectangle(*rect*, ***kwargs*)

Create a Polygon from a rectangle of the kind returned by sirilpy.connection.get_siril_selection().

Parameter

- **rect** (*Tuple[int, int, int, int]*) -- Rectangle as (x, y, width, height)
- ****kwargs** -- Additional keyword arguments to pass to Polygon constructor (polygon_id, color, fill, legend)

Rückgabe
A new Polygon instance representing the rectangle

Rückgabetyp
Polygon

contains_point(*x*, *y*)

Determine if a point is inside the polygon using Dan Sunday's optimized winding number algorithm.

This algorithm is more robust than ray casting for complex polygons and handles edge cases better, including points on edges and self-intersecting polygons.

Parameter

- **x** (*float*) -- X-Koordinate des zu testenden Punktes.
- **y** (*float*) -- Y-Koordinate des zu testenden Punktes.

Rückgabe
True if the point is inside the polygon, False otherwise.

Rückgabetyp
bool

get_bounds()

Get the bounding box of the polygon.

Rückgabe
(min_x, min_y, max_x, max_y)

Rückgabetyp
Tuple[float, float, float, float]

Verursacht
ValueError -- If the polygon has no points.

get_max_x()

Die maximale x-Koordinate des Polygons ermitteln.

Rückgabetyp
float

get_max_y()

Die maximale y-Koordinate des Polygons ermitteln.

Rückgabetyp
`float`

get_min_x()

Die minimale x-Koordinate des Polygons ermitteln.

Rückgabetyp
`float`

get_min_y()

Die minimale y-Koordinate des Polygons ermitteln.

Rückgabetyp
`float`

serialize()

Serializes a single Polygon object into a byte array.

Rückgabe
A byte array representing the serialized polygon data.

Rückgabetyp
bytes

Verursacht
ValueError -- If the number of points exceeds the allowed limit.

color: int = 4294967295

32-bit RGBA color (packed, uint_8 per component. Default value is 0xFFFFFFFF)

fill: bool = False

whether or not the polygon should be filled when drawn

legend: str = None

an optional legend

points: List[*FPoint*]

List of points defining the polygon's shape

polygon_id: int = 0

unique identifier

class sirilpy.models.**RegData**(*fwhm=0.0*, *weighted_fwhm=0.0*, *roundness=0.0*, *quality=0.0*, *background_lvl=0.0*, *number_of_stars=0*, *H=<factory>*)

Bases: `object`

Python equivalent of Siril regdata structure

classmethod deserialize(*data*)

Deserialize a binary response into a RegData object.

Parameter
data (*bytes*) -- Binary data to unpack

Rückgabetyp
RegData

Returns: RegData object

Raises: SirilError if the received data doesn't match the expected size'
struct.error If unpacking fails

H: *Homography*

Stores a homography matrix describing the affine transform from this frame to the reference frame

background_lvl: float32 = 0.0

Hintergrundwert

fwhm: float = 0.0

copy of fwhm->fwhmx, used as quality indicator

number_of_stars: int = 0

die Anzahl der im Bild erkannten Sterne

quality: float = 0.0

measure of image quality

roundness: float32 = 0.0

fwhm->fwhmy / fwhm->fwhmx, 0 when uninit,]0, 1] when set

weighted_fwhm: float32 = 0.0

used to exclude spurious images

class sirilpy.models.**Sequence**(*seqname=''*, *number=0*, *selnum=0*, *fixed=0*, *nb_layers=-1*, *rx=0*, *ry=0*, *is_variable=False*, *bitpix=0*, *reference_image=0*, *imgparam=None*, *regparam=None*, *stats=None*, *distoparam=None*, *beg=0*, *end=0*, *exposure=0.0*, *fz=False*, *type=None*, *cfa_opened_monochrome=False*, *current=0*)

Bases: `object`

Python equivalent of Siril sequ structure

beg: int = 0

imgparam[0]->filenum

bitpix: int = 0

image pixel format, from fits

cfa_opened_monochrome: bool = False

CFA SER opened in monochrome mode

current: int = 0

file number currently loaded

distoparam: List[*DistoData*] = None

distortion data for the sequence [nb_layers]

end: int = 0

imgparam[number-1]->filenum

exposure: float = 0.0

exposure of frames

fixed: int = 0

fixed length of image index in filename

fz: bool = False

ob die Datei komprimiert ist

imgparam: List[*ImgData*] = None

eine Struktur für jedes Bild der Sequenz [Nummer]

is_variable: bool = False

sequence has images of different sizes

nb_layers: int = -1

number of layers embedded in each image file

number: int = 0

Anzahl der Bilder in der Sequenz

reference_image: int = 0

Referenzbild für die Registrierung

regparam: List[List[*RegData*]] = None

registration parameters for each layer [nb_layers][number]

rx: int = 0

first image width

ry: int = 0

first image height

selnum: int = 0

Anzahl ausgewählter Bilder

seqname: str = ''

Name der Sequenz

stats: List[List[*ImageStats*]] = None

statistics of the images for each layer [nb_layers][number]

type: *SequenceType* = None

Typ der Sequenz

25.5 Sirilpy Enums

This module provides all the enums that are used within sirilpy. Enums submodule for Siril. This submodule contains all the enums used within sirilpy.

class sirilpy.enums.**BitpixType**(*value*)

Bases: IntEnum

Mimics the Siril bitpix enum. Note that although Siril can handle opening FITS files of any data type, internally it processes images only as USHORT_IMG (uint16) or FLOAT_IMG (float32).

BYTE_IMG = 8

DOUBLE_IMG = -64

FLOAT_IMG = -32

LONG_IMG = 32

SHORT_IMG = 16

USHORT_IMG = 20

class sirilpy.enums.**CommandStatus**(*value*)

Bases: IntEnum

Contains Siril command status codes, matching the values returned internally within Siril. These can be used for error handling. CMD_OK and CMD_NO_WAIT are no-error codes; all the other codes represent command errors. These are available through the CommandError exception and may generally be handled without being regarded as fatal to the script.

CMD_ALLOC_ERROR = 1048576

CMD_ARG_ERROR = 32

CMD_DIR_NOT_FOUND = 4194304

CMD_FILE_NOT_FOUND = 131072

CMD_FOR_CFA_IMAGE = 65536

CMD_FOR_PLATE_SOLVED = 262144

CMD_GENERIC_ERROR = 128

CMD_IMAGE_NOT_FOUND = 256

CMD_INVALID_IMAGE = 1024

CMD_LOAD_IMAGE_FIRST = 2048

CMD_NEED_INIT_FIRST = 524288

CMD_NOT_FOR_MONO = 16384

CMD_NOT_FOR_RGB = 32768

CMD_NOT_FOR_SINGLE = 8192

CMD_NOT_FOUND = 1

CMD_NOT_SCRIPTABLE = 8

CMD_NO_CWD = 4

CMD_NO_WAIT = 2

CMD_OK = 0

CMD_ONLY_SINGLE_IMAGE = 4096

CMD_SELECTION_ERROR = 64

CMD_SEQUENCE_NOT_FOUND = 512

CMD_THREAD_RUNNING = 2097152

CMD_WRONG_N_ARG = 16

class sirilpy.enums.**DialogID**(*value*)

Bases: IntEnum

classmethod from_name(*name*)

Safely create from a str containing a potential dialog name (e.g. "ASINH_DIALOG", returning None if the name no longer exists.

Rückgabetyp
DialogID | None

ABERRATION_DIALOG = 19

ABOUT_DIALOG = 0

ANNOTATE_DIALOG = 1

ASINH_DIALOG = 2

ASTROMETRY_DIALOG = 3

BACKGROUND_EXTRACTION_DIALOG = 4

BINXY_DIALOG = 6

CANON_FIXBANDING_DIALOG = 7

CCM_DIALOG = 8

CLAHE_DIALOG = 9

COLOR_CALIBRATION = 10

COMPOSITION_DIALOG = 12

COMPSTARS_DIALOG = 11

COSMETIC_DIALOG = 13

CURVES_DIALOG = 14

CWD_DIALOG = 15

DECONV_DIALOG = 5

DENOISE_DIALOG = 16

DIALOG_FFT = 17

DIALOG_STAR_REMIX = 18

EPF_DIALOG = 20

EXTRACT_CHANNEL_DIALOG = 21

EXTRACT_WAVELETS_LAYERS_DIALOG = 22

FILE_INFORMATION = 23

GHT_DIALOG = 24

HISTOGRAM_DIALOG = 25

ICC_DIALOG = 26

KEYWORDS_DIALOG = 27

LINEARMATCH_DIALOG = 28

MEDIAN_DIALOG = 29

MERGE_CFA_DIALOG = 30

NINA_LIGHT_CURVE = 31

OPEN_DIALOG = 32

PCC_DIALOG = 33

PIXEL_MATH_DIALOG = 34

PREFS_DIALOG = 35

RESAMPLE_DIALOG = 36

RGRADIENT_DIALOG = 37

ROTATION_DIALOG = 38

SATU_DIALOG = 40

SAVEAS_DIALOG = 41

SCNR_DIALOG = 42

SEQLIST_DIALOG = 43

SPLIT_CFA_DIALOG = 44

STARNET_DIALOG = 45

STARS_LIST_WINDOW = 46

STAT_WINDOW = 47

S_PCC_DIALOG = 39

UNPURPLE_DIALOG = 48

WAVELETS_DIALOG = 49

class sirilpy.enums.**DialogReq**(*value*)

Bases: IntEnum

Represents the requirements to open a dialog represented by a DialogID

ANY = 1

Image or sequence loaded

IMG = 2

Image loaded

MONO = 3

Mono image loaded

NONE = 4

No requirement, this action works at any time

PLTSOLVD = 5

Image or sequence frame loaded and plate solved

RGB = 6

RGB image loaded

RGBPLTSOLVD = 7

RGB image or sequence frame loaded and plate solved

SEQ = 8

Sequence loaded

class sirilpy.enums.**DistoType**(*value*)

Bases: IntEnum

Python equivalent of the Siril disto_source enum

DISTO_FILE = 2

Distortion from given file

DISTO_FILES = 4

Distortion stored in each file (true only from seq platesolve, even with no distortion, it will be checked upon reloading)

DISTO_FILE_COMET = 5

special for cometary alignement, to be detected by apply reg

DISTO_IMAGE = 1

Verzerrung gegenüber dem aktuellen Bild

DISTO_MASTER = 3

Verzerrung durch Masterdateien

DISTO_UNDEF = 0

keine Verzerrungen

class sirilpy.enums.**ImageType**(*value*)

Bases: IntEnum

enum representing image functional types

BIAS = 4

DARK = 2

FLAT = 3

LIGHT = 1

UNKNOWN = 0

class sirilpy.enums.**LogColor**(*value*)

Bases: IntEnum

Defines colors available for use with `SirilInterface.log()` For consistency `LogColor.Default` should be used for normal messages, `LogColor.Red` should be used for error messages, `LogColor.Salmon` should be used for warning messages, LogColor.Green should be used for completion notifications, and `LogColor.Blue` should be used for technical messages such as equations, coefficients etc.

BLUE = 4

DEFAULT = 0

GREEN = 3

RED = 1

SALMON = 2

class sirilpy.enums.**PlotType**(*value*)

Bases: IntEnum

Enumeration of available plot types for visualizing data series.

HYPHENS = 2

LINES = 3

LINESHYPHENS = 6

LINESMARKS = 5

LINESPOINTS = 4

MARKS = 1

POINTS = 0

class sirilpy.enums.**STFType**(*value*)

Bases: IntEnum

enum representing STF types

ASINH_DISPLAY = 4

AUTOSTRETCH_DISPLAY = 5

HISTEQ_DISPLAY = 6

LINEAR_DISPLAY = 0

LOG_DISPLAY = 1

SQRT_DISPLAY = 2

SQUARED_DISPLAY = 3

class sirilpy.enums.**SequenceType**(*value*)

Bases: IntEnum

Python equivalent of the Siril sequence_type enum

SEQ_AVI = 3

SEQ_FITSEQ = 2

SEQ_INTERNAL = 4

SEQ_REGULAR = 0

SEQ_SER = 1

class sirilpy.enums.**SirilVport**

Bases: `object`

Defines the Siril viewports

BLUE = 2

GREEN = 1

MONO = 0

RED = 0

RGB = 3

class sirilpy.enums.**SlidersMode**(*value*)

Bases: `IntEnum`

MINMAX = 1

MIPSLOHI = 0

USER = 2

class sirilpy.enums.**StarProfile**(*value*)

Bases: `IntEnum`

Python equivalent of the Siril starprofile enum. Used to identify the type of fit used to model a star in the image. Note that MOFFAT_FIXED is currently not used in Siril, but is reserved for future use for Moffat stars modelled with a fixed beta parameter

GAUSSIAN = 0

MOFFAT = 1

MOFFAT_FIXED = 2

25.6 Sirilpy Plotting

This submodule provides classes and method to access the native Siril plotting functionality. Of course, you can also use matplotlib but this submodule provides access to the same plotting capabilities as used internally within Siril for a more seamless result. All its members are made available at the module root level, there is no need to import models separately.

Once populated, the PlotData object can be plotted using `SirilInterface.xy_plot()`. Plot submodule for Siril, providing classes for plot data representation and serialization. This submodule enables users to create and configure various types of plots with customizable appearance and error bars.

class sirilpy.plot.**PlotData**(*title='Data Plot'*, *xlabel='X'*, *ylabel='Y'*, *savename='plot'*, *show_legend=True*, *datamin=None*, *datamax=None*)

Bases: `object`

Metadata container for plot configuration. The actual series data are held in SeriesData objects and can be added using the Class methods add_series or add_series_obj after initialization of the PlotData.

Members:

title: Plot title xlabel: X-axis label ylabel: Y-axis label savename: Save filename (extension is added automatically) show_legend: bool indicating whether to show legend datamin: List [xmin, ymin] forcing the bottom left coordinate to show. If omitted, the range is set to the data range. datamax: List [xmax, ymax] forcing the top right coordinate to show. If omitted, the range is set to the data range.

classmethod serialize(*plot_data*)

Serialize plot data for shared memory transfer using network byte order.

Parameter
plot_data (*PlotData*) -- PlotData object containing plot configuration

Rückgabetyp
`Tuple[bytes, int]`

Rückgabe
Tuple of serialized bytes and total length

add_series(*x_coords*, *y_coords*, *label=None*, *plot_type=PlotType.LINES*, *n_error=None*, *p_error=None*)

Add a new series to the plot metadata.

Rückgabe
the created SeriesData object for further manipulation if needed.

Rückgabetyp
SeriesData

add_series_obj(*series*)

Add a pre-created SeriesData object to the plot metadata.

Returns: None

Rückgabetyp
`None`

class sirilpy.plot.**SeriesData**(*x_coords*, *y_coords*, *label=None*, *plot_type=PlotType.LINES*, *n_error=None*, *p_error=None*)

Bases: `object`

Represents a single data series for plotting.

Members:

x_coords: Either a List[float] or a np.ndarray containing the values for the x coordinates for this series

y_coords: Either a List[float] or a np.ndarray containing the values for the y coordinates for this series

label: A str containing a label for the series (shown in the plot legend)

plot_type: a PlotType setting the type of marks to use

n_error: Either a List[float] or a np.ndarray containing values for the y-axis negative errors for this series

p_error: Either a List[float] or a np.ndarray containing values for the y-axis positive errors for this series

25.7 Sirilpy Utility Methoden

This submodule provides utility methods for use in Siril python scripts. The most important is `ensure_installed()` but there are also methods such as `download_with_progress()` which provides a highly robust method for downloading large files with good error recovery through a retries and resume mechanism. Utility module for Siril Python interface providing helper functions for file operations, package management, and standard I/O control to support Siril's scripting capabilities.

class sirilpy.utility.**SuppressedStderr**

Bases: object

This context manager allows suppression of the script's stderr, which can be useful if you are using module functions that are known to produce warnings that you want to avoid distracting the user with, such as FutureWarnings of features that have become deprecated but are in a dependency rather than your own code. The class should be used sparingly and should **not** be used to hide evidence of bad code.

class sirilpy.utility.**SuppressedStdout**

Bases: object

This context manager allows suppression of the script's stdout, which can be useful to avoid flooding the log with stdout messages from an excessively verbose module used in the script.

Beispiel

```
import sirilpy as s
siril = s.SirilInterface()
print("This message will appear in the Siril log")
with s.SuppressedStdout():
    print("This message will not appear")
print("This message will appear again")
```

sirilpy.utility.**check_module_version**(*requires=None*)

Check the version of the Siril module is sufficient to support the script. This is not mandatory if you are only using classes, methods etc. that are provided in the initial public release, but if you rely on methods that are noted int he API documentation as having been added at a particular version of the module then you must check the running sirilpy module supports your script by calling this function.

Siril will not show scripts with unsatisfied check_module_version() calls in the repository list in 'Get Scripts', so if you want a function that enables different code paths for different versions you should use *needs_module_version()* instead.

Parameter

requires (*str*) -- A version format specifier string following the same format used by pip, i.e. it may contain '==1.2', '!=3.4', '>5.6', '>=7.8', or a combination such as '>=1.2,<3.4'

Rückgabe

True if requires = None or if the available sirilpy module version satisfies the version specifier, otherwise False

Verursacht

ValueError -- if requires is an invalid version specifier.

sirilpy.utility.**download_with_progress**(*siril*, *url*, *file_path*, *max_retries=3*, *retry_delay=5*, *resume=True*)

Robust file download method with native Siril progress tracking and error handling using retries and a resume mechanism.

Rückgabetyp

bool

sirilpy.utility.**ensure_installed**(**packages*, *version_constraints=None*, *reinstall=False*)

Ensures that the specified package(s) are installed and meet optional version constraints.

Parameter

- ***packages** (*str or List[str]*) -- Name(s) of the package(s) to ensure are installed.
- **version_constraints** (*str or List[str], optional*) -- Version constraint string(s) (e.g. ">=1.5", "==2.0"). Can be a single constraint or a list matching packages.

- **reinstall** (*bool, optional*) -- Forces reinstallation. Defaults to False.

Rückgabe
True if all packages are successfully installed or already meet constraints.

Rückgabetyp
bool

Verursacht

- *SirilError* -- If package installation fails.
- **ValueError** -- If a different number of constraints is provided to the number of packages to be installed.
- **TimeoutError** -- If pip fails with an apparent timeout.
- **SystemExit** -- If a package that is already imported needs to be reinstalled with a different version.

`sirilpy.utility.`**`human_readable_size`**(*bytes_size*)

Convert bytes to human-readable format.

Parameter
bytes_size (*int*) -- Size in bytes

Rückgabe
Formatted size with appropriate unit (B, KB, MB, GB, TB)

Rückgabetyp
str

Verursacht
TypeError -- on incorrect input type

`sirilpy.utility.`**`needs_module_version`**(*requires=None*)

Check the version of the Siril module is sufficient to support a feature. This allows writing optional code paths for different sirilpy API levels. The function works the same as check_module_version() but the presence of an unsatisfied needs_module_version() call will not prevent a script from showing up in the list in 'Get Scripts'

Parameter
requires (*str*) -- A version format specifier string following the same format used by pip, i.e. it may contain '==1.2', '!=3.4', '>5.6', '>=7.8', or a combination such as '>=1.2,<3.4'

Rückgabe
True if requires = None or if the available sirilpy module version satisfies the version specifier, otherwise False

Verursacht
ValueError -- if requires is an invalid version specifier.

`sirilpy.utility.`**`parse_fits_header`**(*header_text*, *include_comments=False*)

Parse FITS header from text content into a dictionary with support for HIERARCH and CONTINUE keywords.

Handles Siril's newline-separated FITS header format and converts it to a dictionary compatible with astropy.wcs.WCS. Supports extended FITS keywords via HIERARCH and long string values via CONTINUE.

Parameter

- **header_text** (str) -- Raw header string with header cards separated by newlines. Expected format: "KEYWORD = value / comment"
- **include_comments** (bool) -- If True, includes COMMENT and HISTORY cards in the output. If False (default), these cards are skipped.

Rückgabetyp
`Dict[str, Union[str, int, float, bool]]`

Rückgabe
Dictionary mapping FITS header keywords (str) to their parsed values. Values are converted to appropriate Python types: - 'T'/'F' -> bool - Quoted strings -> str (quotes removed, with CONTINUE support for long strings) - Numeric strings -> int or float - Everything else -> str - COMMENT/HISTORY cards -> str (content after keyword, if include_comments=True) - HIERARCH keywords -> str (full hierarchical keyword preserved)

Hinweise

- Filters out warning messages, tracebacks, and other non-header content
- By default ignores COMMENT and HISTORY cards (set include_comments=True to include)
- Skips malformed cards or invalid keywords
- Standard keywords must be 8 characters and alphanumeric (plus underscore/hyphen)
- HIERARCH keywords can be longer and contain spaces/dots
- CONTINUE cards are automatically merged with the previous string value
- Comments after '/' are ignored for regular cards

Beispiel

```
>>> header_str = '''SIMPLE  = T / file conforms to FITS standard
... BITPIX  = -32 / bits per pixel
... HIERARCH ESO DET CHIP1 NAME = 'CCD #1' / detector name
... LONGSTR = 'This is a very long string that needs to be '
... CONTINUE  'continued on the next line'
... COMMENT Test comment'''
>>> result = parse_fits_header(header_str)
>>> result['SIMPLE']
True
>>> result['BITPIX']
-32
>>> result['HIERARCH ESO DET CHIP1 NAME']
'CCD #1'
>>> result['LONGSTR']
'This is a very long string that needs to be continued on the next line'
```

`sirilpy.utility.`**`safe_subprocess_Popen`**(*command*, ***kwargs*)

Provides a safe alternative to subprocess.Popen, accounting for the fact that a subprocess command line may be run from a flatpak environment and require use of flatpak-spawn to ensure it runs outside the sandbox, avoiding issues with subprocesses that rely on environment variables and so on. safe_subprocess_Popen is a drop-in replacement for subprocess.Popen for use in Siril scripts.

Introduced in sirilpy version 1.0.3

`sirilpy.utility.`**`safe_subprocess_run`**(*command*, ***kwargs*)

Provides a safe alternative to subprocess.Popen, accounting for the fact that a subprocess command line may be run from a flatpak environment and require use of flatpak-spawn to ensure it runs outside the sandbox, avoiding issues with subprocesses that rely on environment variables and so on. safe_subprocess_run is a drop-in replacement for subprocess.run for use in Siril scripts.

Introduced in sirilpy version 1.0.3

`sirilpy.utility.truncate_utf8(data, max_bytes)`

Truncates utf8 input. Accepts either bytes or str as input and returns data in the same format as the input.

Parameter
data (*bytes or str*) -- The data to be truncated

Rückgabe
The truncated data

Rückgabetyp
bytes or str

Verursacht
TypeError -- if the input was not bytes or str

`sirilpy.utility.uninstall_package(package_name)`

Uninstall a package using pip, streaming output to stdout. Introduced in sirilpy 1.0.15.

Parameter
package_name (*str*) -- Name of the package to uninstall.

Verursacht
subprocess.CalledProcessError -- If pip uninstallation fails.

25.8 Sirilpy GPU Helfer

This submodule provides helper classes to make it easier to manage GPU framework packages such as ONNX, Torch etc. for use in Siril python scripts. The landscape of these frameworks' support for different GPU architectures on different OSes is rapidly developing and the helpers aim to suggest reliable packages / configurations. This means that the proposed configurations may in some cases be conservative: the aim is to provide a good and robust level of GPU support for as many users as possible in an automated python environment rather than the absolute best, but perhaps fragile, optimisation. GPU helper module for Siril Python interface providing helper functions for detection, installation and testing of GPU-related modules. Initial scope is ONNX, torch and jax

class `sirilpy.gpuhelper.JaxHelper`

Bases: `object`

A helper class for detecting, installing, and testing JAX with appropriate hardware acceleration.

This class automatically detects the system configuration and installs the correct JAX variant (CPU, CUDA, ROCm, etc.) based on available hardware.

ensure_jax()

Wrapper for install_jax() that only installs it if needed, with negligible overhead if it is already installed.

Rückgabetyp
`bool`

get_recommended_backend()

Determine the recommended JAX backend and installation parameters. When multiple GPUs are present, uses priority system to choose the best one.

JAX has limited platform support: - NVIDIA: CUDA on Linux (Windows support experimental/limited) - AMD ROCm: Linux only - Apple Silicon: Metal backend - Intel: Experimental plugin with dependency issues

Rückgabetyp
`Dict[str, Any]`

Rückgabe
Dict with 'backend', 'packages', 'extra_index_url' keys

install_jax(*force_variant=None*, *version_constraint=None*, *force_reinstall=False*)

Install JAX with the appropriate variant for the detected hardware. Use this instead of ensure_installed() to ensure that jax is installed correctly for the given hardware / OS

Parameter

- **force_variant** (Optional[str]) -- Override auto-detection with specific variant (e.g., 'jax[cpu]')
- **version_constraint** (Optional[str]) -- Version constraint string (e.g., '>=0.4.0')

force_reinstall: Forces a reinstallation

Rückgabe
True if installation succeeded, False otherwise

Rückgabetyp
bool

is_jax_installed()

Check if PyTorch is installed without importing it.

Rückgabetyp
bool

status()

Prints the current status of the Jax Helper class in regard to its support for different OSes, GPUs. The world of heterogenous computing is developing rapidly and while support for some of the frameworks for which helpers are available is not yet universally available, hopefully it will improve in the future.

test_jax()

Test JAX functionality and return execution provider.

Rückgabe
the bool returned is True if jax works or False if it does not, and the str is "gpu" if JAX is using GPU, "cpu" if using CPU or None if

Rückgabetyp
Tuple[bool,str]

Verursacht

- **RuntimeError** -- If JAX computation fails or accuracy check fails
- **ImportError** -- If JAX is not installed

uninstall_jax(*dry_run=False*)

Detect and uninstall any existing JAX-related packages.

This is useful when you need to clean up a problematic JAX installation before installing a different variant (e.g., falling back from GPU to CPU).

Parameter
dry_run (bool) -- If True, only detect packages without uninstalling them

Rückgabetyp
Dict[str, Any]

Rückgabe
Dict containing information about detected and uninstalled packages

class sirilpy.gpuhelper.**ONNXHelper**

Bases: object

A class to handle detection and installation of the appropriate ONNX Runtime package based on the system hardware and configuration.

Example usage (this should be used instead of `sirilpy.ensure_installed("onnxruntime")` to install the correct package for the user's system.)

```
oh = sirilpy.ONNXHelper()
oh.ensure_onnxruntime()
```

ensure_onnxruntime()

Wrapper for install_onnxruntime() that only installs it if needed, with negligible overhead if it is already installed.

Rückgabetyp
bool

get_execution_providers_ordered(*ai_gpu_acceleration=True*, *force_check=False*)

Get execution providers ordered by priority. This function returns a list of available ONNX Runtime execution providers in a reasonable order of priority, covering major GPU platforms: The CPU provider is always included as the final fallback option.

Parameter

- **ai_gpu_acceleration** (*bool*) -- Whether to include GPU acceleration providers. Defaults to True.
- **force_check** (*bool*) -- Whether to force re-checking even if a cached config exists. Defaults to False.

Rückgabe
Ordered list of available execution providers.

Rückgabetyp
list

get_recommended_backend()

Determine the recommended ONNX Runtime backend based on hardware. When multiple GPUs are present, uses priority system to choose the best one.

Note: ONNX Runtime has platform-specific backend support: - Windows: DirectML for all GPUs (NVIDIA, AMD, Intel) - more reliable, no driver dependencies - Linux: CUDA for NVIDIA, CPU for AMD (no ROCm support yet), OpenVINO for Intel

Rückgabe
'gpu', 'directml', 'openvino', 'coreml', or 'cpu'

Rückgabetyp
Backend name

import_onnxruntime()

Import onnxruntime, add it to the global dict, test if it's built against CUDA and if so preload the CUDA and CUDNN libraries to improve the chances of finding them if Torch[CUDA] happens to be installed.

install_onnxruntime(*force=False*)

Detect system configuration and install the appropriate ONNX Runtime package.

Parameter
force -- bool: If True, force reinstallation even if onnxruntime is already installed.

Rückgabe
True if installation was successful or already installed, False otherwise.

Rückgabetyp
bool

Verursacht
`TimooutError` -- if a TimeoutError occurs in ensure_installed() - this avoids falling back to the CPU-only package purely because of network issues

`is_onnxruntime_installed()`

Check if any onnxruntime package is already installed and usable.

Rückgabe

(is_installed, package_name) where package_name could be
'onnxruntime', 'onnxruntime-gpu', 'onnxruntime-silicon', etc.

Rückgabetyp
tuple

`run(session, model_path, output_names, input_feed, run_options=None, return_first_output=False)`

Run inference with automatic CPU fallback if the session fails.

Parameter

- `session` -- The ONNX runtime inference session
- `model_path` (*str*) -- Path to the ONNX model file (needed for CPU fallback)
- `output_names` -- Names of the outputs to compute, or None for all outputs
- `input_feed` -- Dictionary mapping input names to input tensors
- `run_options` -- Optional run options for the inference session
- `return_first_output` (*bool*) -- If True, return only the first output instead of the full list

Rückgabe

(result, session) where result is the inference output (or first output if return_first_output=True) and
session is the (potentially updated) inference session

Rückgabetyp
tuple

`status()`

Prints the current status of the ONNX Helper class in regard to its support for different OSes, GPUs. The world of heterogenous computing is developing rapidly and while support for some of the frameworks for which helpers are available is not yet universally available, hopefully it will improve in the future.

`test_onnxruntime(ort=None)`

Test an imported onnxruntime.

Args:install_torch(cuda_version=cuda_version)

ort: The ONNX runtime module to test. If None, the method will
attempt to import onnxruntime for the test.

Rückgabe
a list of confirmed working ONNXRuntime ExecutionProviders in priority order

Rückgabetyp
list

`uninstall_onnxruntime()`

Detects and uninstalls all variants of onnxruntime packages. Checks for any package starting with 'onnxruntime'.

Rückgabe
A list of uninstalled packages

Rückgabetyp
list

`class sirilpy.gpuhelper.TorchHelper`

Bases: `object`

Helper class for PyTorch detection, installation and testing.

`ensure_torch(cuda_version=None)`

Ensure PyTorch is installed with the appropriate backend.

Parameter
cuda_version (`Optional[str]`) -- Optional CUDA version to override auto-detection (e.g., 'cu118', 'cu126', 'cu128')

Rückgabetyp
`bool`

Returns: True on success, False on failure

`get_recommended_backend()`

Determine the recommended PyTorch backend and installation parameters. When multiple GPUs are present, uses priority system to choose the best one.

PyTorch has good cross-platform support: - NVIDIA: CUDA on all platforms - AMD ROCm-compatible: ROCm on Linux AND Windows - Intel Arc: XPU backend - Apple Silicon: MPS backend

Rückgabetyp
`Dict[str, Any]`

Rückgabe
Dict with 'backend', 'cuda_version', 'extra_index_url', 'packages' keys

`get_torch_device(use_gpu=True)`

Obtains a suitable torch device based on the capabilities of the installed torch package. if use_gpu is False, torch.device("cpu") will be returned. This function is available since sirilpy 1.0.17

Rückgabetyp
torch.device

`install_torch(cuda_version=None, extra_index_url=None, packages=None)`

Install PyTorch with specified configuration.

Parameter

- **cuda_version** (`Optional[str]`) -- CUDA version (e.g., 'cu118', 'cu126', 'cu128')
- **extra_index_url** (`Optional[str]`) -- PyTorch wheel repository URL
- **packages** (`Optional[list]`) -- List of packages to install

is_torch_installed()

Check if PyTorch is installed without importing it.

Rückgabetyp
bool

status()

Prints the current status of the Torch Helper class in regard to its support for different OSes, GPUs. The world of heterogenous computing is developing rapidly and while support for some of the frameworks for which helpers are available is not yet universally available, hopefully it will improve in the future.

test_torch()

Run tests to verify that torch is installed and runs correctly.

uninstall_torch()

Detects and uninstalls PyTorch and related packages. Checks for torch ecosystem packages. :returns: A list of uninstalled packages :rtype: list

sirilpy.gpuhelper.**detect_gpu_capabilities()**

Comprehensive GPU detection for all supported hardware.

Rückgabetyp
Dict[str, Any]

Rückgabe
Dict containing detected hardware information and recommendations for PyTorch, ONNX, and JAX backends.

sirilpy.gpuhelper.**get_gpu_detection_report()**

Generate a comprehensive GPU detection report suitable for display in a GUI.

Rückgabetyp
Dict[str, Any]

Rückgabe
Dict containing all detected hardware and recommended configurations for ONNX, PyTorch, and JAX, including GPU priority information.

25.9 Sirilpy TK GUI

Warnung

This submodule is now deprecated. There are no longer any scripts in the repository that use tksiril and submissions to the siril-scripts repository will no longer be accepted if they use tksiril. You should migrate any personal scripts to use PyQt6 instead. **The tksiril submodule will be removed in a future version of Siril.**

This submodule provides some helper functions to support consistent GUI implementation using `tkinter`. It must be explicitly imported in order to be used. Note that when writing TKinter GUIs you should import ThemedTK from the ttkthemes module, because the bare TKinter UI shows up poorly on MacOS. ThemedTK and the methods available in the `tksiril` module help to provide a consistent look for Siril script GUIs on all platforms.

Warnung

Linux users running a Wayland desktop should note that Tkinter does not yet have support for pure Wayland. In order to use python scripts utilising Tkinter GUIs you **must** have the XWayland compatibility package installed. If you don't, you will see errors about DISPLAY being unavailable.

```
from sirilpy import tksiril
```

TKsiril submodule for Siril, providing utility methods to achieve consistent script GUI appearance using the TKinter toolkit.

DEPRECATION WARNING: the tksiril submodule is deprecated from Siril 1.4.3. It will remain part of sirilpy throughout the stable 1.4 series but has been removed from master and will therefore disappear with the release of Siril 1.6.0.

No new scripts will be accepted in the siril-scripts repository if they use the tksiril submodule.

class sirilpy.tksiril.**ScrollableFrame**(*container*, **args*, ***kwargs*)

Bases: `Frame`

A scrollable frame widget that can contain other widgets.

This class creates a frame with vertical scrolling capability using a Canvas widget and Scrollbar. It supports both scrollbar and mouse wheel scrolling across all platforms (Windows, Mac, Linux).

Usage:

scrollable = ScrollableFrame(parent) scrollable.pack(fill="both", expand=True)

Add widgets to scrollable.scrollable_frame label = ttk.Label(scrollable.scrollable_frame, text="Hello") label.pack()

Optionally bind mouse wheel to child widgets scrollable.add_mousewheel_binding(label)

add_mousewheel_binding(*widget=None*)

Add mouse wheel scrolling support to a widget and its children.

This method recursively binds mouse wheel events to the specified widget and all its child widgets. It uses platform detection to apply the appropriate event bindings for each operating system.

Parameter

widget -- The tkinter widget to bind mouse wheel events to. The binding will be applied recursively to all children. If no widget is specified it will default to the ScrollableFrame itself.

Beispiel

Add a complex widget to the scrollable frame frame = ttk.Frame(scrollable.scrollable_frame) label = ttk.Label(frame, text="Hello") button = ttk.Button(frame, text="Click me")

Bind mouse wheel to the entire widget hierarchy scrollable.add_mousewheel_binding(frame)

sirilpy.tksiril.**create_tooltip**(*widget*, *text*, *wrap_length=250*)

Create a tooltip for a given Tkinter widget.

Parameter

- **widget** (*tk.Widget*) -- The widget to attach the tooltip to
- **text** (*str*) -- The tooltip text to display
- **max_width** (*int, optional*) -- Maximum width of the tooltip. Defaults to 300.
- **wrap_length** (*int, optional*) -- Length at which text wraps. Defaults to 250.

Verursacht
TypeError -- If text is not a string or the provided widget is not a valid Tkinter widget

`sirilpy.tksiril.elevate(root)`

Raises the Tk root window to the top of the window stack. Useful after calls to sirilpy methods that present child windows of the main Siril window such as info_messagebox().

NOTE: For this to work on KDE desktops, focus-stealing prevention must be disabled.

`sirilpy.tksiril.match_theme_to_siril(themed_tk, s, on_top=False)`

Match the Tkinter theme to the Siril configuration and set the script dialog to have topmost status, meaning that it will remain in front of other non-topmost windows.

Parameter

- **s** (`SirilInterface`) -- sirilpy.SirilInterface class to provide the Siril theme (light or dark) to match
- **themed_tk** (*ThemedTk*) -- ThemedTk instance to apply the theme to
- **on_top** -- whether the script window should be always on top of other windows

Verursacht

- **TypeError** -- If input arguments are of incorrect type
- **ValueError** -- If the theme configuration is not 0 or 1
- **AttributeError** -- If required methods are not available
- **RuntimeError** -- If there are errors installing or setting the theme

`sirilpy.tksiril.standard_style()`

Provide a standardised ttk style to allow consistent visual appearance between different Siril python scripts.

Parameter
none

Verursacht
SirilError -- If the style creation or configuration fails

25.10 Sirilpy Tkfilebrowser

> **Warnung**
>
> **This submodule is now deprecated.** There are no longer any scripts in the repository that use tkfilebrowser and submissions to the siril-scripts repository will no longer be accepted if they use tkfilebrowser. You should migrate any personal scripts to use PyQt6 instead. **The tkfilebrowser submodule will be removed in a future version of Siril.**

This submodule is a fork of tkfilebrowser. The fork addresses a bug in the original code where duplicate device entries could cause errors in generating the filebrowser, and ensures the code can be mantained as the upstream package was last updated several years ago.

Documentation for tkfilebrowser can be found here. Note that some compatibility improvements have been made in the version included in sirilpy:

- `sirilpy.tkfilebrowser.askdirectory()` has been added as an alias for `tkfilebrowser.askopendirname()` to maintain compatibility with `tk.filedialog.askdirectory()`.

- In filefilter specifications `tkfilebrowser` requires multiple extensions to be provided separated by | whereas filedialog requires space-separated extensions: `sirilpy.tkfilebrowser` has been adapted to accept either format.

This submodule serves the sole purpose of being a drop-in replacement for the standard Tk filedialog on Linux, as the standard Tk filedialog is horrible on Linux. It can be used as a replacement like this:

```
if sys.platform.startswith("linux"):
    import sirilpy.tkfilebrowser as filedialog
else:
    from tkinter import filedialog
```

25.11 Sirilpy Exceptions

This submodule provides some customized exceptions to support Siril-specific error handling. All its members are made available at the module root level, there is no need to import it separately.

Die Sirilpy-Ausnahmeregelung lautet wie folgt:

- At a low level within sirilpy methods a variety of exception types may be raised (those shown below as well as exceptions raised from other modules such as struct.error). Internal exception types are all descended from SirilError and may therefore be caught using *except SirilError*. Other exception types are re-raised as a SirilError to show the method where they were generated, but the underlying error can still be seen either in a traceback or using the Exception `__cause__` property.
- Some error types are recoverable errors such as NoImageError, NoSequenceError and CommandError. These exception types can be handled at script level (for example by showing a warning dialog reminding the user to load an image).
- Other error types are typically not recoverable, such as SharedMemoryError or SirilConnectionError.

Exceptions submodule for Siril, providing exception classes for use in exception raising within the sirilpy module.

exception `sirilpy.exceptions.`**`CommandError`**(*message='Command execution failed'*, *status_code=CommandStatus.CMD_GENERIC_ERROR*)

Bases: *SirilError*

Raised when a command sent to Siril fails to execute properly. (Note: 'command' in this case refers to internal commands sent from the python module to the Siril python handler, not Siril commands of the type that might be entered in the Siril command entry.) The full set of command status codes is shown in the CommandStatus enum. These exceptions are often recoverable and should therefore be handled before generically handling other SirilError types that are considered fatal.

status_code

(CommandStatus) Indicates the status code returned by the Siril command. This may be used in error handlers to allow scripts to handle some types of command error and continue (e.g. by prompting a user intervention).

exception `sirilpy.exceptions.`**`DataError`**(*message='Error handling data'*)

Bases: *SirilError*

Raised when there are problems with data handling. This includes cases like:

- Invalid image data
- Datenkonvertierungsfehler
- Fehler bei der Speicherzuweisung
- Buffer overflows

exception sirilpy.exceptions.**ImageDialogOpenError**(*message='Siril image dialog is open'*)

Bases: *SirilError*

Exception raised when an image processing dialog is open.

exception sirilpy.exceptions.**MouseModeError**(*message='Siril mouse mode error'*)

Bases: *SirilError*

Exception raised when Siril is in the wrong mouse mode.

exception sirilpy.exceptions.**NoImageError**(*message='No Siril image loaded'*)

Bases: *SirilError*

Raised when a method requires an image to be loaded but no image is loaded. These exceptions are often recoverable and should therefore be handled before generically handling other SirilError types that are considered fatal.

exception sirilpy.exceptions.**NoSequenceError**(*message='No Siril sequence loaded'*)

Bases: *SirilError*

Raised when a method requires a sequence to be loaded but no sequence is loaded. These exceptions are often recoverable and should therefore be handled before generically handling other SirilError types that are considered fatal.

exception sirilpy.exceptions.**ProcessingThreadBusyError**(*message='Siril processing thread is busy'*)

Bases: *SirilError*

Exception raised when the processing thread is already in use.

exception sirilpy.exceptions.**SharedMemoryError**(*message='Siril shared memory error'*)

Bases: *SirilError*

Wird ausgelöst, wenn Probleme bei der Verbindung oder Kommunikation mit Siril über den gemeinsamen Speicher auftreten.

SharedMemoryError is not raised directly but will be wrapped in a SirilError. It should generally be regarded as fatal and the script should shut down gracefully if possible or just stop.

exception sirilpy.exceptions.**SirilConnectionError**(*message='Failed to connect to Siril'*)

Bases: *SirilError*

Raised when there are problems connecting to or communicating with Siril. This includes cases like:

- Siril ist nicht gestartet
- Socket connection failures
- Communication protocol errors
- Unexpected disconnections

SirilConnectionError is not raised directly but will be wrapped in a SirilError. It should generally be regarded as fatal and the script should shut down gracefully if possible or just stop.

exception sirilpy.exceptions.**SirilError**(*message='An error occurred'*)

Bases: `Exception`

Base exception class for all Siril-related errors.

All other Siril exceptions inherit from this class, making it easy to catch any Siril-related error with a single except clause.

KAPITEL 26

Wie man Probleme meldet

Wenn Sie trotz des Studiums der Dokumentation und der Tutorials auf ein seltsames Verhalten stoßen, finden Sie auf dieser Seite Hinweise, was Sie in einer solchen Situation tun können. Denken Sie daran, dass es nicht unbedingt einen Fehler gibt, wenn etwas nicht so funktioniert wie Sie es erwarten. In den meisten Fällen entstehen Probleme durch falsche Verwendung der Software. In solchen Fällen ist es wichtig, genaue Informationen zum Problem bereitzustellen, damit wir feststellen können, ob es sich um einen Fehler oder ein Missverständnis handelt. Weitere Informationen finden Sie im Abschnitt „So senden Sie uns nützliche Informationen“ (Probleme: Senden Sie uns nützliche Informationen). Wenn Sie uns nur sagen, dass es nicht funktioniert, hilft uns das nicht, eine Lösung zu finden. Wir sind keine Wahrsager und benötigen Informationen, um das Problem zu identifizieren und eine Lösung zu entwickeln. Daher ist es wichtig, dass Sie uns mitteilen, was Sie getan haben, als das Problem auftrat, welches Betriebssystem Sie verwenden, welche Version von Siril Sie verwenden und vor allem Protokolle erstellen!

26.1 Änderungsprotokolle und Bug-Tracker prüfen

Zunächst einmal, wenn Sie einen Fehler in Siril finden, kann es sein, dass er bereits gemeldet wurde (manchmal buchstäblich Dutzende von Malen). Wir bitten Sie daher, zuerst zu prüfen, z.B. indem Sie sich changelogs, oder tickets ansehen, die bereits **geöffnet** und sogar **geschlossen** wurden.

26.2 Senden Sie uns nützliche Informationen

Wie in der Einleitung erwähnt, benötigen wir zusätzliche Informationen, um das Problem zu lösen:

1. Welches Betriebssystem verwenden Sie? Da sich Siril unter Windows, Linux oder macOS sehr unterschiedlich verhalten kann, benötigen wir diese Information. Bitte seien Sie so genau wie möglich.
2. Welche Version von Siril verwenden Sie? Und wie haben Sie es bekommen? Das Paket wurde von der Siril-Webseite heruntergeladen? Über einen Drittanbieter? Selber kompiliert? Bitte seien Sie auch hier so präzise wie möglich.
3. Manchmal ist es nützlich, Bildschirmfotos zu teilen. Bitte machen Sie jedoch **keine Screenshots mit Ihrem Smartphone** - sie sind unleserlich. Ihr Betriebssystem ist in der Lage, ganz einfach Screenshots zu erstellen (Google <https://www.google.com>`_ kann Ihnen dabei helfen) und auch Siril bietet eine solche Funktion (die

Taste mit der Kamera). Schließlich sind die gewünschten Formate Bildformate: `jpg`, `bmp` oder `png`, aber absolut nicht `pdf`.

4. Schicken Sie uns Protokolle. Idealerweise bevorzugen wir englische Protokolle! Gehen Sie einfach zu den Siril-Einstellungen und ändern Sie die Sprache auf Englisch in der Registerkarte Benutzeroberfläche. Außerdem gibt es zwei Arten von Protokollen: die in der Siril-Konsole angezeigten, die die von der Software ausgeführten Schritte beschreiben und uns bei der Fehlersuche helfen können, und die internen Protokolle, die sichtbar sind, wenn Siril über die Befehlszeile ausgeführt wird:

   - Erstere sind in den meisten Fällen sehr nützlich und können ganz einfach über die Schaltfläche unten rechts exportiert werden. Dadurch wird eine Datei erstellt, die Sie uns einfach zusenden können.
   - Wenn die Software jedoch abstürzt (d. h. plötzlich ohne Vorwarnung beendet wird), müssen Sie Siril über die Befehlszeile starten und versuchen, den Absturz zu reproduzieren und die Protokolle abzurufen. Hier hängt die Methode vom Betriebssystem ab.

     – Microsoft Windows: Öffnen Sie ein Kommandofenster (geben Sie `cmd` in der Windows-Suchleiste ein) und geben Sie Folgendes ein:

       ```
       "C:\Program Files\Siril\bin\siril.exe" 2>&1 >output.log
       ```

       Dies speichert die Datei output.log in dem Ordner, in dem das Terminal gestartet wurde (in den meisten Fällen im Ordner `%USERPROFILE%`).

     – macOS: Wenn Sie Siril im Ordner "Programme" installiert haben, was im Allgemeinen empfohlen wird, öffnen Sie zunächst die Anwendung Terminal im Ordner "Dienstprogramme" in "Programme" und fügen Sie die folgende Zeile ein:

       ```
       /Applications/Siril.app/Contents/MacOS/siril > ~/Desktop/output.log 2>&1
       ```

       Nach dem Absturz sind die Protokolle auf dem Desktop in der Datei `output.log` verfügbar.

     – GNU/Linux: Starten Sie Siril einfach in einem Terminal. Normalerweise befindet sich die Binärdatei in der `$PATH`-Variable, in diesem Fall geben Sie ein:

       ```
       siril > output.log 2>&1
       ```

       ist alles, was Sie brauchen, um die Protokolle in die Datei `output.log` umzuleiten. Dies speichert die Datei output.log in dem Ordner, in dem das Terminal gestartet wurde

5. Schicken Sie uns Ihr Bild. Wenn Sie Ihr Bild seltsam finden, zögern Sie nicht, es mit uns zu teilen, normalerweise im FITS-Format. Das ist immer interessanter als ein Bildschirmfoto. Dazu müssen Sie einen Dienst für den Austausch großer Dateien nutzen. Es gibt viele davon, und wir können zum Beispiel WeTransfer empfehlen. In diesem Fall laden Sie Ihre Daten zu WeTransfer hoch und erhalten einen Download-Link zum Weitergeben.

26.3 Wie können Sie uns kontaktieren?

Es gibt mehrere Möglichkeiten, uns zu kontaktieren und einen Fehler zu melden. Am einfachsten ist es, uns im Forum zu finden. Aber es ist auch möglich, ein Ticket in unserem gitlab repository zu eröffnen. In diesem Fall überprüfen Sie bitte zuerst, ob das gleiche Ticket nicht schon einmal geöffnet wurde. In diesem Fall wird eine kurze Beschreibung mit einer Ticketnummer im Changelog angezeigt. Um das Ticket (auch ein geschlossenes) zu sehen und zu bestätigen, dass Sie dasselbe Problem haben oder nicht, gehen Sie an die Adresse https://gitlab.com/free-astro/siril/-/issues/XXXX mit XXXX als Ticketnummer.

Tipp

Da Siril-Benutzer und -Entwickler aus unterschiedlichen Ländern kommen, müssen Fehlerberichte auf englisch gemeldet werden.

Literaturverzeichnis

[Fischler1981] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

[Valdes1995] Valdes, F. G., Campusano, L. E., Velasquez, J. D., & Stetson, P. B. (1995). FOCAS automatic catalog matching algorithms. Publications of the Astronomical Society of the Pacific, 107(717), 1119.

[Losson2010] Olivier Losson, Ludovic Macaire, Yanqin Yang. Comparison of color demosaicing methods. Advances in Imaging and Electron Physics, 2010, 162, pp.173-265, section 2.2.2. https://hal.science/hal-00683233/document

[FruchterHook1997] A. S. Fruchter and R. N. Hook. (1997) *A novel image reconstruction method applied to deep Hubble Space Telescope images.* Proc. S.P.I.E. vol. 3164. https://arxiv.org/abs/astro-ph/9708242

[Avila2015] R. J. Avila, A. Koekemoer, J. Mack, A. Fruchter. (2015) *Optimizing pixfrac in Astrodrizzle: An Example from the Hubble Frontier Fields.* https://www.stsci.edu/files/live/sites/www/files/home/hst/instrumentation/wfc3/documentation/instrument-science-reports-isrs/_documents/2015/WFC3-2015-04.pdf

[Peter2009] Peter J. Huber und E. Ronchetti (2009), Robust Statistics, 2te Ed., Wiley

[ConejeroPI] Juan Conejero, ImageIntegration, Pixinsight Tutorial

[Rosner1983] Rosner, B. (1983). *Percentage points for a generalized ESD many-outlier procedure*. Technometrics, 25(2), 165-172.

[Brown2007] Brown, M., & Lowe, D. G. (2007). *Automatic panoramic image stitching using invariant features*. International journal of computer vision, 74, 59-73.

[GaiaDR3] Vallenari, A., et al. "Gaia Data Release 3-Summary of the content and survey properties." Astronomy & Astrophysics 674 (2023): A1. 99(613), 191.

[SWIRE] https://www.iasf-milano.inaf.it/~polletta/templates/swire_templates.html

[Stellar] https://www.eso.org/sci/facilities/paranal/decommissioned/isaac/tools/lib.html

[Wiki_HEALPIX] https://en.wikipedia.org/wiki/HEALPix

[Young1994] https://opg.optica.org/ao/viewmedia.cfm?uri=ao-33-6-1108&seq=0

[Anger2018] Anger, J., Facciolo, G., & Delbracio, M. (2018). *Estimating an image's blur kernel using natural image statistics, and deblurring it: an analysis of the Goldstein-Fattal method.* Image Processing On Line, 8, 282-304. https://doi.org/10.5201/ipol.2018.211

[Anger2019] Anger, J., Facciolo, G., & Delbracio, M. (2019). Blind image deblurring using the l0 gradient prior. Image processing on line, 9, 124-142. https://doi.org/10.5201/ipol.2019.243

[Goldstein2012] Goldstein, A., & Fattal, R. (2012, October). *Blur-kernel estimation from spectral irregularities.* In European Conference on Computer Vision (pp. 622-635). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33715-4_45

[Lucy1974] Lucy, L. B. (1974). *An iterative technique for the rectification of observed distributions.* The astronomical journal, 79, 745. https://doi.org/10.1086/111605.

[Lebrun2013] Lebrun, M., Buades, A., & Morel, J. M. (2013) *Implementation of the "Non-Local Bayes" (NL-Bayes) Image Denoising Algorithm.* Image Processing On Line, 3 , pp. 1–42. https://doi.org/10.5201/ipol.2013.16

[Pierazzo2017] Pierazzo, N., & Facciolo, G. (2017). *Data adaptive dual domain denoising: a method to boost state of the art denoising algorithms.* Image Processing On Line, 7, 93-114. https://doi.org/10.5201/ipol.2017.203

[Mäkitalo2011] Mäkitalo, M., & Foi, A. (2012, March). *Poisson-gaussian denoising using the exact unbiased inverse of the generalized anscombe transformation.* In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1081-1084). IEEE. https://doi.org/10.1109/ICASSP.2012.6288074

[Mäkitalo2012] Makitalo, M., & Foi, A. (2011). *A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation.* IEEE transactions on image processing, 20(9), 2697-2698. https://doi.org/10.1109/TIP.2011.2121085

[Romano2015] Romano, Y., & Elad, M. (2015). *Boosting of image denoising algorithms.* SIAM Journal on Imaging Sciences, 8(2), 1187-1219. https://doi.org/10.1137/140990978

[Wright2003] Wright, Grady Barrett. Radial basis function interpolation: numerical and analytical developments. University of Colorado at Boulder, 2003.

[Stetson1987] Stetson, P. B. (1987). DAOPHOT: A computer program for crowded-field stellar photometry. Publikation der Astronomical Society of the Pacific, 99(613), 191.

Python-Modulindex

S

sirilpy.connection, 577
sirilpy.enums, 619
sirilpy.exceptions, 638
sirilpy.gpuhelper, 630
sirilpy.models, 602
sirilpy.plot, 625
sirilpy.tksiril, 636
sirilpy.utility, 626

Commands Index

A

addmax, 501
asinh, 502
autoghs, 502
autostretch, 502

B

bg, 503
bgnoise, 503
binxy, 503
boxselect, 503

C

calibrate, 504
calibrate_single, 504
capabilities, 505
catsearch, 505
ccm, 506
cd, 506
cdg, 506
clahe, 507
clear, 507
clearstar, 507
close, 507
conesearch, 508
convert, 508
convertraw, 509
cosme, 509
cosme_cfa, 510
crop, 510

D

ddp, 510
denoise, 511
dir, 512
disto, 512
dumpheader, 512

E

entropy, 512
epf, 513
exit, 513
extract, 513
extract_Green, 514
extract_Ha, 514
extract_HaOIII, 514

F

fdiv, 514
ffill, 515
fftd, 515
ffti, 515
fill, 515
find_cosme, 516
find_cosme_cfa, 516
find_hot, 516
findcompstars, 517
findstar, 517
fix_xtrans, 518
fixbanding, 518
fmedian, 518
fmul, 519

G

gauss, 519
get, 519
getref, 519
ght, 520
grey_flat, 520

H

help, 520
histo, 521

I

iadd, 521
icc_assign, 521
icc_convert_to, 522
icc_remove, 522

idiv, 522
imul, 522
inspector, 523
invght, 523
invmodasinh, 523
invmtf, 524
isub, 524

J

jsonmetadata, 524

L

light_curve, 524
limit, 525
linear_match, 525
link, 526
linstretch, 526
livestack, 526
load, 527
load_seq, 527
log, 527
ls, 528

M

makepsf, 528
merge, 529
merge_cfa, 529
mirrorx, 530
mirrorx_single, 530
mirrory, 530
modasinh, 530
mtf, 531

N

neg, 531
new, 532
nozero, 532

O

offline, 532
offset, 532
online, 533

P

parse, 533
pcc, 533
platesolve, 534
pm, 535
profile, 535
psf, 536
pwd, 536
pyscript, 536

R

register, 537
reloadscripts, 538
requires, 538
resample, 539
rgbcomp, 539
rgradient, 539
rl, 540
rmgreen, 540
rotate, 541
rotatePi, 541

S

satu, 541
save, 542
savebmp, 542
savejpg, 542
savejxl, 543
savepng, 543
savepnm, 543
savetif, 543
savetif32, 544
savetif8, 544
sb, 544
select, 545
seqapplyreg, 545
seqccm, 547
seqclean, 547
seqcosme, 547
seqcosme_cfa, 548
seqcrop, 548
seqextract_Green, 548
seqextract_Ha, 549
seqextract_HaOIII, 549
seqfind_cosme, 549
seqfind_cosme_cfa, 550
seqfindstar, 550
seqfixbanding, 550
seqght, 551
seqheader, 551
seqinvght, 551
seqinvmodasinh, 551
seqlinstretch, 552
seqmerge_cfa, 552
seqmodasinh, 552
seqmtf, 553
seqprofile, 553
seqpsf, 553
seqplatesolve, 554
seqresample, 555
seqrl, 556
seqsb, 556
seqsetmag, 556
seqsplit_cfa, 557

seqstarnet, 557
seqstat, 557
seqsubsky, 558
seqtilt, 558
sequnsetmag, 558
sequpdate_key, 559
seqwiener, 559
set, 559
set16bits, 560
set32bits, 560
setcompress, 560
setcpu, 560
setext, 561
setfindstar, 561
setmag, 562
setmem, 562
setphot, 563
setref, 563
show, 563
spcc, 564
spcc_list, 565
split, 565
split_cfa, 565
stack, 566
stackall, 567
starnet, 568
start_ls, 568
stat, 569
stop_ls, 569
subsky, 569
synthstar, 570

T

threshlo, 570
threshhi, 570
thresh, 570
tilt, 571
trixel, 571

U

unclipstars, 571
unpurple, 572
unselect, 572
unsetmag, 572
unsharp, 573
update_key, 573

V

visu, 573

W

wavelet, 573
wiener, 574
wrecons, 574

Stichwortverzeichnis

A

A (*Attribut von sirilpy.models.PSFStar*), 613
A_err (*Attribut von sirilpy.models.PSFStar*), 613
ABERRATION_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
ABOUT_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
add_mousewheel_binding() (*Methode von sirilpy.tksiril.ScrollableFrame*), 636
add_series() (*Methode von sirilpy.plot.PlotData*), 626
add_series_obj() (*Methode von sirilpy.plot.PlotData*), 626
addmax, **501**
airmass (*Attribut von sirilpy.models.FKeywords*), 606
airmass (*Attribut von sirilpy.models.ImgData*), 612
allocate_data() (*Methode von sirilpy.models.FFit*), 604
analyse_image_from_file() (*Methode von sirilpy.connection.SirilInterface*), 577
ang_err (*Attribut von sirilpy.models.PSFStar*), 613
angle (*Attribut von sirilpy.models.PSFStar*), 614
ANNOTATE_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
ANY (*Attribut von sirilpy.enums.DialogReq*), 622
aperture (*Attribut von sirilpy.models.FKeywords*), 606
asinh, **502**
ASINH_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
ASINH_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
ASTROMETRY_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
autoghs, **502**
autostretch, **502**
AUTOSTRETCH_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
avgDev (*Attribut von sirilpy.models.ImageStats*), 611

B

B (*Attribut von sirilpy.models.PSFStar*), 613
B_err (*Attribut von sirilpy.models.PSFStar*), 613
BACKGROUND_EXTRACTION_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
background_lvl (*Attribut von sirilpy.models.RegData*), 618
bayer_pattern (*Attribut von sirilpy.models.FKeywords*), 606
bayer_xoffset (*Attribut von sirilpy.models.FKeywords*), 606
bayer_yoffset (*Attribut von sirilpy.models.FKeywords*), 606
beg (*Attribut von sirilpy.models.Sequence*), 618
beta (*Attribut von sirilpy.models.PSFStar*), 614
beta_err (*Attribut von sirilpy.models.PSFStar*), 614
bg, **503**
bgnoise, **503**
bgnoise (*Attribut von sirilpy.models.ImageAnalysis*), 610
bgnoise (*Attribut von sirilpy.models.ImageStats*), 611
BGSample (*Klasse in sirilpy.models*), 602
BIAS (*Attribut von sirilpy.enums.ImageType*), 623
binning_x (*Attribut von sirilpy.models.FKeywords*), 606
binning_y (*Attribut von sirilpy.models.FKeywords*), 606
binxy, **503**
BINXY_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
bitpix (*Attribut von sirilpy.models.FFit*), 604
bitpix (*Attribut von sirilpy.models.Sequence*), 618
BitpixType (*Klasse in sirilpy.enums*), 619
BLUE (*Attribut von sirilpy.enums.LogColor*), 623
BLUE (*Attribut von sirilpy.enums.SirilVport*), 625
Bmag (*Attribut von sirilpy.models.PSFStar*), 613
boxselect, **503**
bscale (*Attribut von sirilpy.models.FKeywords*), 606
BV (*Attribut von sirilpy.models.PSFStar*), 613
BYTE_IMG (*Attribut von sirilpy.enums.BitpixType*), 619
bzero (*Attribut von sirilpy.models.FKeywords*), 606

C

calibrate, **504**
calibrate_single, **504**
CANON_FIXBANDING_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
capabilities, **505**
catsearch, **505**
ccd_temp (*Attribut von sirilpy.models.FKeywords*), 607
ccm, **506**
CCM_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
cd, **506**
cdg, **506**
centalt (*Attribut von sirilpy.models.FKeywords*), 607
centaz (*Attribut von sirilpy.models.FKeywords*), 607
cfa_opened_monochrome (*Attribut von sirilpy.models.Sequence*), 618
channels (*Attribut von sirilpy.models.ImageAnalysis*), 610
channels (*sirilpy.models.FFit property*), 604
check_module_version() (*im Modul sirilpy.utility*), 627
checksum (*Attribut von sirilpy.models.FFit*), 605
clahe, **507**
CLAHE_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
clear, **507**
clear_image_bgsamples() (*Methode von sirilpy.connection.SirilInterface*), 577
clear_undo_history() (*Methode von sirilpy.connection.SirilInterface*), 577
clearstar, **507**
close, **507**
cmd() (*Methode von sirilpy.connection.SirilInterface*), 577
CMD_ALLOC_ERROR (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_ARG_ERROR (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_DIR_NOT_FOUND (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_FILE_NOT_FOUND (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_FOR_CFA_IMAGE (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_FOR_PLATE_SOLVED (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_GENERIC_ERROR (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_IMAGE_NOT_FOUND (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_INVALID_IMAGE (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_LOAD_IMAGE_FIRST (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NEED_INIT_FIRST (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NO_CWD (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NO_WAIT (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NOT_FOR_MONO (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NOT_FOR_RGB (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NOT_FOR_SINGLE (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NOT_FOUND (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_NOT_SCRIPTABLE (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_OK (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_ONLY_SINGLE_IMAGE (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_SELECTION_ERROR (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_SEQUENCE_NOT_FOUND (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_THREAD_RUNNING (*Attribut von sirilpy.enums.CommandStatus*), 620
CMD_WRONG_N_ARG (*Attribut von sirilpy.enums.CommandStatus*), 620
color (*Attribut von sirilpy.models.Polygon*), 617
COLOR_CALIBRATION (*Attribut von sirilpy.enums.DialogID*), 621
color_managed (*Attribut von sirilpy.models.FFit*), 605
command_error_message() (*Methode von sirilpy.connection.SirilInterface*), 578
CommandError, 638
CommandStatus (*Klasse in sirilpy.enums*), 619
COMPOSITION_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
COMPSTARS_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
conesearch, **508**
confirm_messagebox() (*Methode von sirilpy.connection.SirilInterface*), 578
connect() (*Methode von sirilpy.connection.SirilInterface*), 578
contains_point() (*Methode von sirilpy.models.Polygon*), 616
convert, **508**
convertraw, **509**
cosme, **509**
cosme_cfa, **510**
COSMETIC_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
create_new_seq() (*Methode von sirilpy.connection.SirilInterface*), 578

create_tooltip() (*im Modul sirilpy.tksiril*), 636
crop, **510**
current (*Attribut von sirilpy.models.Sequence*), 618
CURVES_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
cvf (*Attribut von sirilpy.models.FKeywords*), 607
CWD_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621

D

DARK (*Attribut von sirilpy.enums.ImageType*), 623
data (*sirilpy.models.FFit property*), 605
data_max (*Attribut von sirilpy.models.FKeywords*), 607
data_min (*Attribut von sirilpy.models.FKeywords*), 607
DataError, 638
date (*Attribut von sirilpy.models.FKeywords*), 607
date_obs (*Attribut von sirilpy.models.FKeywords*), 607
date_obs (*Attribut von sirilpy.models.ImgData*), 612
ddp, **510**
dec (*Attribut von sirilpy.models.FKeywords*), 607
dec (*Attribut von sirilpy.models.PSFStar*), 614
DECONV_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
DEFAULT (*Attribut von sirilpy.enums.LogColor*), 624
denoise, **511**
DENOISE_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
deserialize() (*Klassenmethode von sirilpy.models.BGSample*), 603
deserialize() (*Klassenmethode von sirilpy.models.FKeywords*), 606
deserialize() (*Klassenmethode von sirilpy.models.ImageAnalysis*), 610
deserialize() (*Klassenmethode von sirilpy.models.ImageStats*), 611
deserialize() (*Klassenmethode von sirilpy.models.ImgData*), 612
deserialize() (*Klassenmethode von sirilpy.models.PSFStar*), 613
deserialize() (*Klassenmethode von sirilpy.models.RegData*), 617
deserialize_polygon() (*Klassenmethode von sirilpy.models.Polygon*), 615
deserialize_polygon_list() (*Klassenmethode von sirilpy.models.Polygon*), 615
detect_gpu_capabilities() (*im Modul sirilpy.gpuhelper*), 635
DIALOG_FFT (*Attribut von sirilpy.enums.DialogID*), 621
DIALOG_STAR_REMIX (*Attribut von sirilpy.enums.DialogID*), 621
DialogID (*Klasse in sirilpy.enums*), 620
DialogReq (*Klasse in sirilpy.enums*), 622
dir, **512**
disconnect() (*Methode von sirilpy.connection.SirilInterface*), 579
disto, **512**
DISTO_FILE (*Attribut von sirilpy.enums.DistoType*), 623
DISTO_FILE_COMET (*Attribut von sirilpy.enums.DistoType*), 623
DISTO_FILES (*Attribut von sirilpy.enums.DistoType*), 623
DISTO_IMAGE (*Attribut von sirilpy.enums.DistoType*), 623
DISTO_MASTER (*Attribut von sirilpy.enums.DistoType*), 623
DISTO_UNDEF (*Attribut von sirilpy.enums.DistoType*), 623
DistoData (*Klasse in sirilpy.models*), 603
distoparam (*Attribut von sirilpy.models.Sequence*), 618
DistoType (*Klasse in sirilpy.enums*), 623
DOUBLE_IMG (*Attribut von sirilpy.enums.BitpixType*), 619
download_with_progress() (*im Modul sirilpy.utility*), 627
dumpheader, **512**

E

elevate() (*im Modul sirilpy.tksiril*), 637
end (*Attribut von sirilpy.models.Sequence*), 618
ensure_data_type() (*Methode von sirilpy.models.FFit*), 604
ensure_installed() (*im Modul sirilpy.utility*), 627
ensure_jax() (*Methode von sirilpy.gpuhelper.JaxHelper*), 630
ensure_onnxruntime() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 632
ensure_torch() (*Methode von sirilpy.gpuhelper.TorchHelper*), 634
entropy, **512**
epf, **513**
EPF_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
error_messagebox() (*Methode von sirilpy.connection.SirilInterface*), 579
estimate_noise() (*Methode von sirilpy.models.FFit*), 604
exit, **513**
expend (*Attribut von sirilpy.models.FKeywords*), 607
exposure (*Attribut von sirilpy.models.FKeywords*), 607
exposure (*Attribut von sirilpy.models.Sequence*), 618
expstart (*Attribut von sirilpy.models.FKeywords*), 607
extract, **513**
EXTRACT_CHANNEL_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
extract_Green, **514**
extract_Ha, **514**
extract_HaOIII, **514**
EXTRACT_WAVELETS_LAYERS_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621

F

fdiv, **514**
ffill, **515**
FFit (*Klasse in sirilpy.models*), 603

fftd, **515**
ffti, **515**
fhi (*Attribut von sirilpy.models.FKeywords*), 607
FILE_INFORMATION (*Attribut von sirilpy.enums.DialogID*), 621
filename (*Attribut von sirilpy.models.DistoData*), 603
filename (*Attribut von sirilpy.models.FKeywords*), 607
filenum (*Attribut von sirilpy.models.ImgData*), 612
fill, **515**
fill (*Attribut von sirilpy.models.Polygon*), 617
filter (*Attribut von sirilpy.models.FKeywords*), 607
filter (*Attribut von sirilpy.models.ImageAnalysis*), 610
find_cosme, **516**
find_cosme_cfa, **516**
find_hot, **516**
findcompstars, **517**
findstar, **517**
fix_xtrans, **518**
fixbanding, **518**
fixed (*Attribut von sirilpy.models.Sequence*), 618
FKeywords (*Klasse in sirilpy.models*), 606
FLAT (*Attribut von sirilpy.enums.ImageType*), 623
FLEN (*Attribut von sirilpy.models.ImageAnalysis*), 610
FLEN_VALUE (*Attribut von sirilpy.models.FKeywords*), 606
flength (*Attribut von sirilpy.models.FKeywords*), 607
flo (*Attribut von sirilpy.models.FKeywords*), 607
FLOAT_IMG (*Attribut von sirilpy.enums.BitpixType*), 619
fmedian, **518**
fmul, **519**
focal_length (*Attribut von sirilpy.models.FKeywords*), 607
focname (*Attribut von sirilpy.models.FKeywords*), 608
foctemp (*Attribut von sirilpy.models.FKeywords*), 608
focuspos (*Attribut von sirilpy.models.FKeywords*), 608
focussz (*Attribut von sirilpy.models.FKeywords*), 608
FPoint (*Klasse in sirilpy.models*), 609
from_name() (*Klassenmethode von sirilpy.enums.DialogID*), 620
from_rectangle() (*Klassenmethode von sirilpy.models.Polygon*), 616
fwhm (*Attribut von sirilpy.models.ImageAnalysis*), 610
fwhm (*Attribut von sirilpy.models.RegData*), 618
fwhmx (*Attribut von sirilpy.models.PSFStar*), 614
fwhmx_arcsec (*Attribut von sirilpy.models.PSFStar*), 614
fwhmy (*Attribut von sirilpy.models.PSFStar*), 614
fwhmy_arcsec (*Attribut von sirilpy.models.PSFStar*), 614
fz (*Attribut von sirilpy.models.Sequence*), 618

G

gain (*Attribut von sirilpy.models.FKeywords*), 608
gauss, **519**
GAUSSIAN (*Attribut von sirilpy.enums.StarProfile*), 625
get, **519**
get_bounds() (*Methode von sirilpy.models.Polygon*), 616
get_channel() (*Methode von sirilpy.models.FFit*), 604
get_execution_providers_ordered() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 632
get_gpu_detection_report() (*im Modul sirilpy.gpuhelper*), 635
get_image() (*Methode von sirilpy.connection.SirilInterface*), 579
get_image_bgsamples() (*Methode von sirilpy.connection.SirilInterface*), 580
get_image_filename() (*Methode von sirilpy.connection.SirilInterface*), 580
get_image_fits_header() (*Methode von sirilpy.connection.SirilInterface*), 580
get_image_history() (*Methode von sirilpy.connection.SirilInterface*), 580
get_image_iccprofile() (*Methode von sirilpy.connection.SirilInterface*), 581
get_image_keywords() (*Methode von sirilpy.connection.SirilInterface*), 581
get_image_pixeldata() (*Methode von sirilpy.connection.SirilInterface*), 581
get_image_shape() (*Methode von sirilpy.connection.SirilInterface*), 582
get_image_stars() (*Methode von sirilpy.connection.SirilInterface*), 582
get_image_stats() (*Methode von sirilpy.connection.SirilInterface*), 582
get_image_unknown_keys() (*Methode von sirilpy.connection.SirilInterface*), 582
get_max_x() (*Methode von sirilpy.models.Polygon*), 616
get_max_y() (*Methode von sirilpy.models.Polygon*), 616
get_min_x() (*Methode von sirilpy.models.Polygon*), 617
get_min_y() (*Methode von sirilpy.models.Polygon*), 617
get_recommended_backend() (*Methode von sirilpy.gpuhelper.JaxHelper*), 630
get_recommended_backend() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 632
get_recommended_backend() (*Methode von sirilpy.gpuhelper.TorchHelper*), 634
get_selection_star() (*Methode von sirilpy.connection.SirilInterface*), 583
get_selection_stats() (*Methode von sirilpy.connection.SirilInterface*), 583
get_seq() (*Methode von sirilpy.connection.SirilInterface*), 584
get_seq_distodata() (*Methode von sirilpy.connection.SirilInterface*), 584
get_seq_frame() (*Methode von sirilpy.connection.SirilInterface*), 584
get_seq_frame_filename() (*Methode von siril-*

py.connection.SirilInterface), 585
get_seq_frame_header() (*Methode von sirilpy.connection.SirilInterface*), 585
get_seq_frame_pixeldata() (*Methode von sirilpy.connection.SirilInterface*), 585
get_seq_imgdata() (*Methode von sirilpy.connection.SirilInterface*), 586
get_seq_regdata() (*Methode von sirilpy.connection.SirilInterface*), 586
get_seq_stats() (*Methode von sirilpy.connection.SirilInterface*), 587
get_siril_active_vport() (*Methode von sirilpy.connection.SirilInterface*), 587
get_siril_config() (*Methode von sirilpy.connection.SirilInterface*), 587
get_siril_configdir() (*Methode von sirilpy.connection.SirilInterface*), 588
get_siril_display_iccprofile() (*Methode von sirilpy.connection.SirilInterface*), 588
get_siril_log() (*Methode von sirilpy.connection.SirilInterface*), 588
get_siril_panzoom() (*Methode von sirilpy.connection.SirilInterface*), 588
get_siril_selection() (*Methode von sirilpy.connection.SirilInterface*), 589
get_siril_slider_state() (*Methode von sirilpy.connection.SirilInterface*), 589
get_siril_stf() (*Methode von sirilpy.connection.SirilInterface*), 589
get_siril_stf_linked() (*Methode von sirilpy.connection.SirilInterface*), 589
get_siril_systemdatadir() (*Methode von sirilpy.connection.SirilInterface*), 589
get_siril_userdatadir() (*Methode von sirilpy.connection.SirilInterface*), 590
get_siril_wd() (*Methode von sirilpy.connection.SirilInterface*), 590
get_torch_device() (*Methode von sirilpy.gpuhelper.TorchHelper*), 634
getref, **519**
ght, **520**
GHT_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
GREEN (*Attribut von sirilpy.enums.LogColor*), 624
GREEN (*Attribut von sirilpy.enums.SirilVport*), 625
grey_flat, **520**

H

H (*Attribut von sirilpy.models.RegData*), 618
h00 (*Attribut von sirilpy.models.Homography*), 609
h01 (*Attribut von sirilpy.models.Homography*), 609
h02 (*Attribut von sirilpy.models.Homography*), 610
h10 (*Attribut von sirilpy.models.Homography*), 610
h11 (*Attribut von sirilpy.models.Homography*), 610
h12 (*Attribut von sirilpy.models.Homography*), 610
h20 (*Attribut von sirilpy.models.Homography*), 610
h21 (*Attribut von sirilpy.models.Homography*), 610
h22 (*Attribut von sirilpy.models.Homography*), 610
has_saturated (*Attribut von sirilpy.models.PSFStar*), 614
header (*Attribut von sirilpy.models.FFit*), 605
height (*Attribut von sirilpy.models.ImageAnalysis*), 610
height (*sirilpy.models.FFit property*), 605
help, **520**
hi (*Attribut von sirilpy.models.FKeywords*), 608
HISTEQ_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
histo, **521**
HISTOGRAM_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
history (*Attribut von sirilpy.models.FFit*), 605
Homography (*Klasse in sirilpy.models*), 609
human_readable_size() (*im Modul sirilpy.utility*), 628
HYPHENS (*Attribut von sirilpy.enums.PlotType*), 624

I

iadd, **521**
icc_assign, **521**
icc_convert_to, **522**
ICC_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
icc_profile (*sirilpy.models.FFit property*), 605
icc_remove, **522**
idiv, **522**
image_lock() (*Methode von sirilpy.connection.SirilInterface*), 590
image_type (*Attribut von sirilpy.models.FKeywords*), 608
ImageAnalysis (*Klasse in sirilpy.models*), 610
ImageDialogOpenError, 638
ImageStats (*Klasse in sirilpy.models*), 611
imagetype (*Attribut von sirilpy.models.ImageAnalysis*), 611
ImageType (*Klasse in sirilpy.enums*), 623
IMG (*Attribut von sirilpy.enums.DialogReq*), 622
ImgData (*Klasse in sirilpy.models*), 612
imgparam (*Attribut von sirilpy.models.Sequence*), 618
import_onnxruntime() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 632
imul, **522**
incl (*Attribut von sirilpy.models.ImgData*), 612
index (*Attribut von sirilpy.models.DistoData*), 603
info_messagebox() (*Methode von sirilpy.connection.SirilInterface*), 591
Inliers (*Attribut von sirilpy.models.Homography*), 609
inspector, **523**
install_jax() (*Methode von sirilpy.gpuhelper.JaxHelper*), 631
install_onnxruntime() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 632

install_torch() (*Methode von sirilpy.gpuhelper.TorchHelper*), 634
instrume (*Attribut von sirilpy.models.FKeywords*), 608
invght, **523**
invmodasinh, **523**
invmtf, **524**
is_cli() (*Methode von sirilpy.connection.SirilInterface*), 591
is_image_loaded() (*Methode von sirilpy.connection.SirilInterface*), 591
is_jax_installed() (*Methode von sirilpy.gpuhelper.JaxHelper*), 631
is_onnxruntime_installed() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 633
is_sequence_loaded() (*Methode von sirilpy.connection.SirilInterface*), 592
is_torch_installed() (*Methode von sirilpy.gpuhelper.TorchHelper*), 634
is_variable (*Attribut von sirilpy.models.Sequence*), 619
iso_speed (*Attribut von sirilpy.models.FKeywords*), 608
isub, **524**

J

JaxHelper (*Klasse in sirilpy.gpuhelper*), 630
jsonmetadata, **524**

K

keywords (*Attribut von sirilpy.models.FFit*), 605
KEYWORDS_DIALOG (*Attribut von sirilpy.enums.DialogID*), 621
KEYWORDS_FORMAT (*Attribut von sirilpy.models.FKeywords*), 606
KEYWORDS_SIZE (*Attribut von sirilpy.models.FKeywords*), 606

L

layer (*Attribut von sirilpy.models.PSFStar*), 614
legend (*Attribut von sirilpy.models.Polygon*), 617
LIGHT (*Attribut von sirilpy.enums.ImageType*), 623
light_curve, **524**
limit, **525**
LINEAR_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
linear_match, **525**
LINEARMATCH_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
LINES (*Attribut von sirilpy.enums.PlotType*), 624
LINESHYPHENS (*Attribut von sirilpy.enums.PlotType*), 624
LINESMARKS (*Attribut von sirilpy.enums.PlotType*), 624
LINESPOINTS (*Attribut von sirilpy.enums.PlotType*), 624
link, **526**
linstretch, **526**
livestack, **526**
livetime (*Attribut von sirilpy.models.FKeywords*), 608
lo (*Attribut von sirilpy.models.FKeywords*), 608
load, **527**
load_image_from_file() (*Methode von sirilpy.connection.SirilInterface*), 592
load_seq, **527**
location (*Attribut von sirilpy.models.ImageStats*), 611
log, **527**
log() (*Methode von sirilpy.connection.SirilInterface*), 592
LOG_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
LogColor (*Klasse in sirilpy.enums*), 623
LONG_IMG (*Attribut von sirilpy.enums.BitpixType*), 619
ls, **528**

M

mad (*Attribut von sirilpy.models.ImageStats*), 611
mag (*Attribut von sirilpy.models.PSFStar*), 614
makepsf, **528**
MARKS (*Attribut von sirilpy.enums.PlotType*), 624
match_theme_to_siril() (*im Modul sirilpy.tksiril*), 637
max (*Attribut von sirilpy.models.BGSample*), 603
max (*Attribut von sirilpy.models.ImageStats*), 611
maxi (*Attribut von sirilpy.models.FFit*), 605
mean (*Attribut von sirilpy.models.BGSample*), 603
mean (*Attribut von sirilpy.models.ImageStats*), 611
median (*Attribut von sirilpy.models.BGSample*), 603
median (*Attribut von sirilpy.models.ImageStats*), 612
MEDIAN_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
merge, **529**
merge_cfa, **529**
MERGE_CFA_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
min (*Attribut von sirilpy.models.BGSample*), 603
min (*Attribut von sirilpy.models.ImageStats*), 612
mini (*Attribut von sirilpy.models.FFit*), 605
MINMAX (*Attribut von sirilpy.enums.SlidersMode*), 625
MIPSLOHI (*Attribut von sirilpy.enums.SlidersMode*), 625
mirrorx, **530**
mirrorx_single, **530**
mirrory, **530**
modasinh, **530**
module
    sirilpy.connection, 577
    sirilpy.enums, 619
    sirilpy.exceptions, 638
    sirilpy.gpuhelper, 630
    sirilpy.models, 602
    sirilpy.plot, 625
    sirilpy.tksiril, 636
    sirilpy.utility, 626

MOFFAT (*Attribut von sirilpy.enums.StarProfile*), 625
MOFFAT_FIXED (*Attribut von sirilpy.enums.StarProfile*), 625
MONO (*Attribut von sirilpy.enums.DialogReq*), 622
MONO (*Attribut von sirilpy.enums.SirilVport*), 625
MouseModeError, 639
mtf, **531**

N

naxes (*sirilpy.models.FFit property*), 605
naxis (*Attribut von sirilpy.models.FFit*), 605
nb_layers (*Attribut von sirilpy.models.Sequence*), 619
nbstars (*Attribut von sirilpy.models.ImageAnalysis*), 611
needs_module_version() (*im Modul sirilpy.utility*), 628
neg, **531**
neg_ratio (*Attribut von sirilpy.models.FFit*), 605
new, **532**
ngoodpix (*Attribut von sirilpy.models.ImageStats*), 612
NINA_LIGHT_CURVE (*Attribut von sirilpy.enums.DialogID*), 622
NoImageError, 639
NONE (*Attribut von sirilpy.enums.DialogReq*), 622
normValue (*Attribut von sirilpy.models.ImageStats*), 612
NoSequenceError, 639
nozero, **532**
number (*Attribut von sirilpy.models.Sequence*), 619
number_of_stars (*Attribut von sirilpy.models.RegData*), 618

O

objctdec (*Attribut von sirilpy.models.FKeywords*), 608
objctra (*Attribut von sirilpy.models.FKeywords*), 608
object (*Attribut von sirilpy.models.FKeywords*), 608
observer (*Attribut von sirilpy.models.FKeywords*), 608
offline, **532**
offset, **532**
offset (*Attribut von sirilpy.models.FKeywords*), 608
online, **533**
ONNXHelper (*Klasse in sirilpy.gpuhelper*), 631
OPEN_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
open_dialog() (*Methode von sirilpy.connection.SirilInterface*), 593
orig_bitpix (*Attribut von sirilpy.models.FFit*), 605
overlay_add_polygon() (*Methode von sirilpy.connection.SirilInterface*), 593
overlay_clear_polygons() (*Methode von sirilpy.connection.SirilInterface*), 593
overlay_delete_polygon() (*Methode von sirilpy.connection.SirilInterface*), 593
overlay_draw_polygon() (*Methode von sirilpy.connection.SirilInterface*), 593
overlay_get_polygon() (*Methode von sirilpy.connection.SirilInterface*), 594
overlay_get_polygons_list() (*Methode von sirilpy.connection.SirilInterface*), 594

P

pair_matched (*Attribut von sirilpy.models.Homography*), 610
parse, **533**
parse_fits_header() (*im Modul sirilpy.utility*), 628
pcc, **533**
PCC_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
phot_is_valid (*Attribut von sirilpy.models.PSFStar*), 614
pix2radec() (*Methode von sirilpy.connection.SirilInterface*), 594
PIXEL_MATH_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
pixel_size_x (*Attribut von sirilpy.models.FKeywords*), 608
pixel_size_y (*Attribut von sirilpy.models.FKeywords*), 608
platesolve, **534**
PlotData (*Klasse in sirilpy.plot*), 625
PlotType (*Klasse in sirilpy.enums*), 624
PLTSOLVD (*Attribut von sirilpy.enums.DialogReq*), 623
pltsolvd (*Attribut von sirilpy.models.FKeywords*), 608
pltsolvd_comment (*Attribut von sirilpy.models.FKeywords*), 609
pm, **535**
POINTS (*Attribut von sirilpy.enums.PlotType*), 624
points (*Attribut von sirilpy.models.Polygon*), 617
Polygon (*Klasse in sirilpy.models*), 615
polygon_id (*Attribut von sirilpy.models.Polygon*), 617
position (*Attribut von sirilpy.models.BGSample*), 603
PREFS_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
ProcessingThreadBusyError, 639
profile, **535**
profile (*Attribut von sirilpy.models.PSFStar*), 614
program (*Attribut von sirilpy.models.FKeywords*), 609
psf, **536**
PSFStar (*Klasse in sirilpy.models*), 613
pwd, **536**
pyscript, **536**

Q

quality (*Attribut von sirilpy.models.RegData*), 618

R

R (*Attribut von sirilpy.models.PSFStar*), 613
ra (*Attribut von sirilpy.models.FKeywords*), 609
ra (*Attribut von sirilpy.models.PSFStar*), 614
radec2pix() (*Methode von sirilpy.connection.SirilInterface*), 595
RED (*Attribut von sirilpy.enums.LogColor*), 624

RED (*Attribut von sirilpy.enums.SirilVport*), 625
redo() (*Methode von sirilpy.connection.SirilInterface*), 595
reference_image (*Attribut von sirilpy.models.Sequence*), 619
RegData (*Klasse in sirilpy.models*), 617
register, **537**
regparam (*Attribut von sirilpy.models.Sequence*), 619
reloadscripts, **538**
requires, **538**
resample, **539**
RESAMPLE_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
reset_progress() (*Methode von sirilpy.connection.SirilInterface*), 595
RGB (*Attribut von sirilpy.enums.DialogReq*), 623
RGB (*Attribut von sirilpy.enums.SirilVport*), 625
rgbcomp, **539**
RGBPLTSOLVD (*Attribut von sirilpy.enums.DialogReq*), 623
rgradient, **539**
RGRADIENT_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
rl, **540**
rmgreen, **540**
rmse (*Attribut von sirilpy.models.PSFStar*), 614
rotate, **541**
rotatePi, **541**
ROTATION_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
roundness (*Attribut von sirilpy.models.ImageAnalysis*), 611
roundness (*Attribut von sirilpy.models.RegData*), 618
row_order (*Attribut von sirilpy.models.FKeywords*), 609
run() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 633
rx (*Attribut von sirilpy.models.ImgData*), 612
rx (*Attribut von sirilpy.models.Sequence*), 619
ry (*Attribut von sirilpy.models.ImgData*), 613
ry (*Attribut von sirilpy.models.Sequence*), 619

S

s_Bmag (*Attribut von sirilpy.models.PSFStar*), 614
s_mag (*Attribut von sirilpy.models.PSFStar*), 614
S_PCC_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
safe_subprocess_Popen() (*im Modul sirilpy.utility*), 629
safe_subprocess_run() (*im Modul sirilpy.utility*), 629
SALMON (*Attribut von sirilpy.enums.LogColor*), 624
sat (*Attribut von sirilpy.models.PSFStar*), 614
satu, **541**
SATU_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
save, **542**
save_image_file() (*Methode von sirilpy.connection.SirilInterface*), 595
SAVEAS_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
savebmp, **542**
savejpg, **542**
savejxl, **543**
savepng, **543**
savepnm, **543**
savetif, **543**
savetif32, **544**
savetif8, **544**
sb, **544**
scale (*Attribut von sirilpy.models.ImageStats*), 612
SCNR_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
ScrollableFrame (*Klasse in sirilpy.tksiril*), 636
select, **545**
selnum (*Attribut von sirilpy.models.Sequence*), 619
SEQ (*Attribut von sirilpy.enums.DialogReq*), 623
SEQ_AVI (*Attribut von sirilpy.enums.SequenceType*), 624
SEQ_FITSEQ (*Attribut von sirilpy.enums.SequenceType*), 624
SEQ_INTERNAL (*Attribut von sirilpy.enums.SequenceType*), 624
SEQ_REGULAR (*Attribut von sirilpy.enums.SequenceType*), 624
SEQ_SER (*Attribut von sirilpy.enums.SequenceType*), 624
seqapplyreg, **545**
seqccm, **547**
seqclean, **547**
seqcosme, **547**
seqcosme_cfa, **548**
seqcrop, **548**
seqextract_Green, **548**
seqextract_Ha, **549**
seqextract_HaOIII, **549**
seqfind_cosme, **549**
seqfind_cosme_cfa, **550**
seqfindstar, **550**
seqfixbanding, **550**
seqght, **551**
seqheader, **551**
seqinvght, **551**
seqinvmodasinh, **551**
seqlinstretch, **552**
SEQLIST_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
seqmerge_cfa, **552**
seqmodasinh, **552**
seqmtf, **553**
seqname (*Attribut von sirilpy.models.Sequence*), 619
seqplatesolve, **554**
seqprofile, **553**
seqpsf, **553**

seqresample, 555
seqrl, 556
seqsb, 556
seqsetmag, 556
seqsplit_cfa, 557
seqstarnet, 557
seqstat, 557
seqsubsky, 558
seqtilt, 558
Sequence (*Klasse in sirilpy.models*), 618
SequenceType (*Klasse in sirilpy.enums*), 624
sequnsetmag, 558
sequpdate_key, 559
seqwiener, 559
serialize() (*Klassenmethode von sirilpy.plot.PlotData*), 626
serialize() (*Methode von sirilpy.models.ImageAnalysis*), 610
serialize() (*Methode von sirilpy.models.Polygon*), 617
SeriesData (*Klasse in sirilpy.plot*), 626
set, 559
set16bits, 560
set32bits, 560
set_image_bgsamples() (*Methode von sirilpy.connection.SirilInterface*), 596
set_image_filename() (*Methode von sirilpy.connection.SirilInterface*), 596
set_image_iccprofile() (*Methode von sirilpy.connection.SirilInterface*), 597
set_image_metadata_from_header_string() (*Methode von sirilpy.connection.SirilInterface*), 597
set_image_pixeldata() (*Methode von sirilpy.connection.SirilInterface*), 598
set_seq_frame_incl() (*Methode von sirilpy.connection.SirilInterface*), 598
set_seq_frame_pixeldata() (*Methode von sirilpy.connection.SirilInterface*), 599
set_siril_pan() (*Methode von sirilpy.connection.SirilInterface*), 599
set_siril_selection() (*Methode von sirilpy.connection.SirilInterface*), 599
set_siril_slider_lohi() (*Methode von sirilpy.connection.SirilInterface*), 600
set_siril_slider_mode() (*Methode von sirilpy.connection.SirilInterface*), 600
set_siril_stf() (*Methode von sirilpy.connection.SirilInterface*), 600
set_siril_stf_linked() (*Methode von sirilpy.connection.SirilInterface*), 601
set_siril_zoom() (*Methode von sirilpy.connection.SirilInterface*), 601
set_temp (*Attribut von sirilpy.models.FKeywords*), 609
setcompress, 560
setcpu, 560
setext, 561
setfindstar, 561
setmag, 562
setmem, 562
setphot, 563
setref, 563
SharedMemoryError, 639
SHORT_IMG (*Attribut von sirilpy.enums.BitpixType*), 619
show, 563
sigma (*Attribut von sirilpy.models.ImageStats*), 612
SirilConnectionError, 639
SirilError, 639
SirilInterface (*Klasse in sirilpy.connection*), 577
sirilpy.connection
    module, 577
sirilpy.enums
    module, 619
sirilpy.exceptions
    module, 638
sirilpy.gpuhelper
    module, 630
sirilpy.models
    module, 602
sirilpy.plot
    module, 625
sirilpy.tksiril
    module, 636
sirilpy.utility
    module, 626
SirilVport (*Klasse in sirilpy.enums*), 625
siteelev (*Attribut von sirilpy.models.FKeywords*), 609
sitelat (*Attribut von sirilpy.models.FKeywords*), 609
sitelat_str (*Attribut von sirilpy.models.FKeywords*), 609
sitelong (*Attribut von sirilpy.models.FKeywords*), 609
sitelong_str (*Attribut von sirilpy.models.FKeywords*), 609
size (*Attribut von sirilpy.models.BGSample*), 603
SlidersMode (*Klasse in sirilpy.enums*), 625
SNR (*Attribut von sirilpy.models.PSFStar*), 613
spcc, 564
spcc_list, 565
split, 565
split_cfa, 565
SPLIT_CFA_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
SQRT_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
sqrtbwmv (*Attribut von sirilpy.models.ImageStats*), 612
SQUARED_DISPLAY (*Attribut von sirilpy.enums.STFType*), 624
stack, 566
stackall, 567
stackcnt (*Attribut von sirilpy.models.FKeywords*), 609

standard_style() (*im Modul sirilpy.tksiril*), 637
star_name (*Attribut von sirilpy.models.PSFStar*), 614
starnet, **568**
STARNET_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
StarProfile (*Klasse in sirilpy.enums*), 625
STARS_LIST_WINDOW (*Attribut von sirilpy.enums.DialogID*), 622
start_ls, **568**
stat, **569**
STAT_WINDOW (*Attribut von sirilpy.enums.DialogID*), 622
stats (*Attribut von sirilpy.models.FFit*), 605
stats (*Attribut von sirilpy.models.Sequence*), 619
status() (*Methode von sirilpy.gpuhelper.JaxHelper*), 631
status() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 633
status() (*Methode von sirilpy.gpuhelper.TorchHelper*), 635
status_code (*Attribut von sirilpy.exceptions.CommandError*), 638
STFType (*Klasse in sirilpy.enums*), 624
stop_ls, **569**
subsky, **569**
SuppressedStderr (*Klasse in sirilpy.utility*), 626
SuppressedStdout (*Klasse in sirilpy.utility*), 627
sx (*Attribut von sirilpy.models.PSFStar*), 614
sx_err (*Attribut von sirilpy.models.PSFStar*), 615
sy (*Attribut von sirilpy.models.PSFStar*), 615
sy_err (*Attribut von sirilpy.models.PSFStar*), 615
synthstar, **570**

T

telescop (*Attribut von sirilpy.models.FKeywords*), 609
test_jax() (*Methode von sirilpy.gpuhelper.JaxHelper*), 631
test_onnxruntime() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 633
test_torch() (*Methode von sirilpy.gpuhelper.TorchHelper*), 635
thresh, **570**
threshhi, **570**
threshlo, **570**
tilt, **571**
timestamp (*Attribut von sirilpy.models.ImageAnalysis*), 611
top_down (*Attribut von sirilpy.models.FFit*), 605
TorchHelper (*Klasse in sirilpy.gpuhelper*), 634
total (*Attribut von sirilpy.models.ImageStats*), 612
trixel, **571**
truncate_utf8() (*im Modul sirilpy.utility*), 629
type (*Attribut von sirilpy.models.Sequence*), 619

U

unclipstars, **571**
undo() (*Methode von sirilpy.connection.SirilInterface*), 601
undo_save_state() (*Methode von sirilpy.connection.SirilInterface*), 601
uninstall_jax() (*Methode von sirilpy.gpuhelper.JaxHelper*), 631
uninstall_onnxruntime() (*Methode von sirilpy.gpuhelper.ONNXHelper*), 634
uninstall_package() (*im Modul sirilpy.utility*), 630
uninstall_torch() (*Methode von sirilpy.gpuhelper.TorchHelper*), 635
units (*Attribut von sirilpy.models.PSFStar*), 615
UNKNOWN (*Attribut von sirilpy.enums.ImageType*), 623
unknown_keys (*Attribut von sirilpy.models.FFit*), 605
unpurple, **572**
UNPURPLE_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
unselect, **572**
unsetmag, **572**
unsharp, **573**
update_key, **573**
update_progress() (*Methode von sirilpy.connection.SirilInterface*), 601
update_stats() (*Methode von sirilpy.models.FFit*), 604
USER (*Attribut von sirilpy.enums.SlidersMode*), 625
USHORT_IMG (*Attribut von sirilpy.enums.BitpixType*), 619

V

valid (*Attribut von sirilpy.models.BGSample*), 603
velocity (*Attribut von sirilpy.models.DistoData*), 603
visu, **573**

W

warning_messagebox() (*Methode von sirilpy.connection.SirilInterface*), 602
wavelet, **573**
WAVELETS_DIALOG (*Attribut von sirilpy.enums.DialogID*), 622
weighted_fwhm (*Attribut von sirilpy.models.RegData*), 618
wfwhm (*Attribut von sirilpy.models.ImageAnalysis*), 611
width (*Attribut von sirilpy.models.ImageAnalysis*), 611
width (*sirilpy.models.FFit property*), 605
wiener, **574**
wrecons, **574**

X

x (*Attribut von sirilpy.models.FPoint*), 609
x0 (*Attribut von sirilpy.models.PSFStar*), 615
x_err (*Attribut von sirilpy.models.PSFStar*), 615
xpos (*Attribut von sirilpy.models.PSFStar*), 615

`xy_plot()` (*Methode von siril-py.connection.SirilInterface*), 602

Y

`y` (*Attribut von sirilpy.models.FPoint*), 609
`y0` (*Attribut von sirilpy.models.PSFStar*), 615
`y_err` (*Attribut von sirilpy.models.PSFStar*), 615
`ypos` (*Attribut von sirilpy.models.PSFStar*), 615