Siril

Выпуск 1.2.0

Free-Astro Team

июл. 24, 2023
1 Установка 3
1.1 Установка в GNU/Linux ................................. 4
1.2 Установка в Microsoft Windows .......................... 5
1.3 Установка в macOS ....................................... 9
1.4 Установка из исходных кодов ............................ 11

2 Графический интерфейс пользователя 15
2.1 Основной интерфейс .................................... 15
2.2 Главное меню ............................................. 23
2.3 Клавиатурные комбинации ............................... 26
2.4 Окно с данными об изображении ........................ 27

3 Настройки 29
3.1 Настройки (графический интерфейс) ........................ 30
3.2 Настройки (команды) .................................... 49

4 Форматы файлов 59
4.1 Разрядность ............................................. 59
4.2 Common File Formats .................................. 62
4.3 FITS ....................................................... 63
4.4 Astro-TIFF ............................................... 66
4.5 SER ....................................................... 70

5 Последовательности 73
5.1 A set of two or more FITS files ........................... 73
5.2 A single SER file ........................................ 74
5.3 A single FITS file ....................................... 74
5.4 Загрузка последовательности ............................ 74
5.5 Frame selector .......................................... 75
5.6 Экспорт последовательности ............................ 77
5.7 Sequence information .................................... 77

6 Определения и рабочий процесс 79

7 Предобработка 81
7.1 Conversion .................................................. 81
7.2 Calibration ................................................ 86
# Обработка

8. Обработка

8.1 Image stretching .......................... 127
8.2 Colors ....................................... 140
8.3 Фильтры ..................................... 150
8.4 Star Processing .............................. 190
8.5 Geometry .................................... 202
8.6 Извлечение фона ............................ 207
8.7 Extraction .................................... 210
8.8 Линейный поиск соответствий ............ 215
8.9 RGB compositing ............................. 216
8.10 Merge CFA Channels ......................... 217
8.11 Pixel Math ................................... 219

# Построение графиков

9. Построение графиков

9.1 Регистрация ............................... 227
9.2 Фотометрия .................................. 231
9.3 Shared options .............................. 233
9.4 Plot interactions ............................ 233

# Динамическая PSF

10. Динамическая PSF

10.1 Первоначальные кандидаты в звёзды ........ 236
10.2 Models ........................................ 237
10.3 Minimization ................................. 241
10.4 Use ........................................... 241
10.5 Configuration ................................ 243
10.6 Литература .................................. 245

# Астрометрия

11. Астрометрия

11.1 Platesolving ................................. 247
11.2 Аннотации .................................. 256

# Фотометрия

12. Фотометрия

12.1 Принципы .................................... 263
12.2 Быстрая фотометрия ......................... 267
12.3 Световые кривые ............................ 274

# Контроль изображений

13. Контроль изображений

13.1 Наклон ....................................... 281
13.2 Аберрации .................................. 283

# Статистика

14. Статистика

14.1 Estimators .................................... 285

# Сценарии

15. Сценарии

15.1 Использование сценариев .................... 289
15.2 Populating the list of scripts ............... 290
15.3 Встроенные сценарии ....................... 292
15.4 Написание собственных сценариев .......... 293

# Автономный режим

16. Автономный режим

16.1 Command Stream (Pipe) ..................... 295
17 Разбор путей 297
  17.1 Syntax example .................................................. 297
  17.2 Finding a file with path parsing .................................. 300
  17.3 Writing a file with path parsing ..................................... 300
  17.4 Using wildcards .................................................. 301

18 Укладка на лету 303
  18.1 Укладка на лету (интерфейс пользователя) ...................... 303
  18.2 Укладка на лету (автономный режим) ........................... 305

19 Команды 307

Литература 369

Алфавитный указатель 371
Siril — это инструмент обработки астрономических изображений, специально разработанный для уменьшения уровня шума и улучшения отношения сигнал/шум изображений, полученных при съёмке большого количества кадров, как это требуется в астрономии.

Язык программирования — C, с частями на C++. Основная разработка ведётся с использованием самых последних версий разделяемых библиотек в GNU/Linux. Желающие внести свой вклад — добро пожаловать.

В документации предпринята попытка описать все функции Siril. Если эквивалент функции из графического интерфейса пользователя существует в командной строке, он указан во врезке. Другие полезные ресурсы можно найти на нашем основном веб-сайте siril.org.

По ссылке команды Siril приведён список доступных в Siril команд.


Установка

Каждая версия Siril доступна на 3 основных платформах — Windows, MacOS, GNU/Linux, и может быть загружена с веб-сайта Siril. Поскольку Siril это свободное программное обеспечение, вы можете скомпилировать его из исходных кодов.

Совет: It may be useful to check the integrity of the binary or package you have just downloaded. The list of SHA keys is available on this page, in json format.

После установки программы, с помощью команды `capabilities` возможно получить больше информации об установленной программе.

Командная строка Siril

```
capabilities
```

Отображает список возможностей Siril на основе данных о компиляции и среде, в которой исполняется программа
3. Порядок версий Siril

Начиная с версии 1.0, стабильные версии Siril (например, 1.0, 1.2) обозначаются чётными номерами и предназначены для повседневного использования. Версии в разработке, которые обозначаются нечётными номерами, например, 0.99.0 или 1.1.0, обычно недоступны в виде пакетов или исполняемых файлов и пользователю необходимо самостоятельно их компилировать. Последняя, третья цифра в номере версии, так называемая минорная версия (например, 1.0.1, 1.0.2, 1.0.3 и пр.), соответствует количеству выпусков, которые включают исправления ошибок и иные небольшие дополнения.

1. Установка в GNU/Linux

1.1 Installation on Debian

The binary package is available on Debian testing and an old version for stable. It can be installed via apt, with superuser privileges:

```
sudo apt install siril
```

1.2 Installation on Ubuntu or Linux Mint

Official repositories

As for debian, it is available in the repositories, but the version can be outdated:

```
sudo apt install siril
```

PPA repositories

The newer version is thus available in our PPA, which is the preferred way to install Siril on Ubuntu or Linux Mint:

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

1.3 Установка через AppImage

For GNU/Linux systems, we also decided to provide bundled binaries with AppImage (x86_64) and flatpak that works on GNU/Linux-like systems. To run the AppImage binary, you just have to download it and allow it to run using the command:

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

Укажите данной команде правильный путь и номер версии (x.y.z) к двоичному файлу AppImage. После этого двойной щелчок по AppImage запускает Siril.
1.1.4 Установка через flatpack

Другим способом установки стабильной версии Siril является использование flatpak, — утилиты для развёртывания и управления пакетами в Linux. Для установки через flatpak введите следующую команду:

```
flatpak install flathub org.free_astro.siril
```

Затем, для запуска приложения:

```
flatpak run org.free_astro.siril
```

1.2 Установка в Microsoft Windows

1.2.1 Установка с помощью инсталлятора

Рекомендуемый способ установки Siril — использовать программу установки, которая проведёт вас шаг за шагом.

Рис. 1: На первом экране программы установки вам необходимо принять соглашение, чтобы продолжить.

Мастер настройки Siril корректно установит все необходимые файлы, и в конце у вас будет выбор создавать или не создавать ярлык на рабочем столе.

Примечание: Siril будет установлен в C:\Program Files\Siril. Если у вас нет прав на установку в эту папку, установите программу в другое место.
Рис. 2: Последний экран программы установки. Вы можете запустить Siril сразу после установки и открыть руководство, объясняющее первые шаги.
паанку, используйте вместо этого переносимую версию (см. Установка переносимой версии.)

1.2.2 Установка переносимой версии

Если вы хотите использовать Siril без установки каких либо файлов на свой компьютер (например, если у вас отсутствуют права администратора на компьютере), то рекомендуется использовать переносимую версию. Она поставляется в виде zip-файла, который вам просто нужно извлечь в папку по вашему выбору, затем перейти в папку bin для запуска siril.exe. Вы также можете создать ярлык на своем рабочем столе, чтобы упростить запуск приложения.

Предупреждение: Будьте остерожны, ни при каких обстоятельствах не перемещайте файл exe или любой другой файл. В противном случае Siril не запустится.

1.2.3 Установка из Microsoft Store

Возможно также установить Siril через Microsoft Store.

1. Перейдите в меню Пуск, где в списке приложений выберите Microsoft Store.
2. Введите Siril в строке поиска.
3. Откройте соответствующую Siril страницу и выберите Получить.

Примечание: Однако необходимо отметить, что обновления Siril в магазине обычно выполняются с некоторой задержкой из-за довольно сложного процесса загрузки.
1.2.4 Установка в Windows с Msys2

Эти инструкции предназначены для компиляции в Windows с дистрибутивом MSYS2 с использованием MinGW. Для MSYS2 требуется 64-разрядная версия Windows 7 или новее и не работает с файловыми системами FAT.

Загрузите MSYS2 64bit, платформу для распространения и компиляции программного обеспечения для Windows, и запустите программу установки x86_64 для 64-разрядной архитектуры. При запросе укажите каталог, куда будет установлена MSYS2 64-bit.

Запустите MSYS2 непосредственно из программы установки или позднее MSYS2 MinGW 64-bit из меню "Пуск" или нажав на ярлык.

Предупреждение: Убедитесь, что запущена MinGW 64-bit (проверьте, что значок в верхней части окна терминала синий).

Сначала обновите базу пакетов и основные системные пакеты набрав (больше информации о pacman на этой странице):

```
pacman -Syu
```

Установка зависимостей

Примечание: Automake это старый (стабильный) метод сборки сейчас заменяется на meson (экспериментальную) систему сборки.

Для установки зависимостей, введите следующую команду:

```
pacman --noconfirm -S --needed base-devel \    
mingw-w64-x86_64-toolchain \    
mingw-w64-x86_64-cmake \    
git \    
automake \    
mingw-w64-x86_64-lcms2 \    
mingw-w64-x86_64-curl \    
mingw-w64-x86_64-json-glib \    
mingw-w64-x86_64-meson \    
mingw-w64-x86_64-ninja \    
mingw-w64-x86_64-fftw \    
mingw-w64-x86_64-exiv2 \    
mingw-w64-x86_64-gtk3 \    
mingw-w64-x86_64-libconfig \    
mingw-w64-x86_64-gsl \    
mingw-w64-x86_64-opencv \    
mingw-w64-x86_64-libheif \    
mingw-w64-x86_64-ffms2 \    
mingw-w64-x86_64-cfitsio \    
mingw-w64-x86_64-libraw
```
Сборка из исходных кодов

Исходный код хранится в репозитории gitlab и в первый раз вы можете скачать его с командой:

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

Теперь создайте систему сборки и скомпилируйте код набрав:

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

Для запуска вашей сборки Siril, запустите MSYS2 64-bit и введите команду:

```
siril
```

Для запуска программы, вы так же можете создать ярлык для siril.exe. Расположение по умолчанию - /mingw64/bin/.

Для обновления вашей версии, запустите MSYS2 64-bit, а потом:

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

Если команда git pull не показывает никаких изменений, то нет необходимости в повторной сборке при помощи команды make. В противном случае ваша сборка будет обновлена.

После этого запустите программу, набрав:

```
siril
```

1.3 Установка в macOS

1.3.1 Установка нашего приложения

Приложение для macOS доступно для:

- Intel (macOS 10.13+)
- Apple Silicon (macOS 11+)

Выберите ссылку, соответствующую архитектуре вашего процессора и скачайте образ диска. После загрузки, дважды кликните по файлу, чтобы открыть его.
Появится новое окно. Сейчас перетащите иконку Siril и сбросьте её на иконку Приложения.

Поздравляем, Siril установлен.

1.3.2 Установка из Homebrew

Homebrew похож на MacPorts и предоставляет пакеты (они же formula) для установки либо путём компиляции их из исходных кодов, либо с использованием предварительно скомпилированных двоичных файлов (они же bottles). Чтобы установить Homebrew, нажмите здесь. Siril может быть установлен с помощью команды:

```
brew install siril
```

Примечание: Пожалуйста, имейте в виду, что Homebrew использует аналитику. Чтобы отключить это, выполните: `brew analytics off`. Подробнее об этом вы можете прочитать в разделе Brew Analytics.
1.4 Установка из исходных кодов

Установка из исходных кодов требуется, если вам нужны новейшие функции, если предыдущая версия устарела, если вы хотите участвовать в улучшении Siril или не хотите использовать все зависимости.

1.4.1 Получение исходных кодов

Исходный код хранится в репозитории git, который в первый раз вы можете загрузить с помощью этой команды:

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

И обновить его в последующем, используя следующие команды в корневом каталоге siril:

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

1.4.2 Зависимости

Siril зависит от ряда библиотек, большинство из которых должны быть доступны в вашей операционной системе, если она достаточно современная. Названия пакетов, специфичных для операционных систем, перечислены в каждом разделе ниже. Обязательными зависимостями являются:

- `gtk+3` (библиотека графического интерфейса пользователя), по крайней мере, версия 3.20.
- `adwaita-icon-theme` (иконки) поддержка внешнего вида приложений `gtk`.
- `cfitsio` (поддержка изображений FITS).
- `fftw` (Библиотека дискретных преобразований Фурье).
- `gsl` (The GNU Scientific Library), версии 1 или 2 начиная с релиза 0.9.1 или ревизии SVN 1040.
- OpenCV и компилятор C++ для некоторых операций с изображениями.

**Примечание:** Даже если Siril может запускаться в консоль начиная с версии 0.9.9, он по-прежнему компонуется с графическими библиотеками, поэтому вам всё равно понадобится GTK+ для компиляции и запуска программы.

Необязательные зависимости:

- `openmp` поддержка мультипотоковости. Хотя эта зависимость опциональна, её использование настоятельно рекомендуется, поскольку это существенно улучшает производительность приложения. По умолчанию эта опция включена. Это означает, что если openmp не установлена на вашей машине, необходимо добавить `-Dopenmp=false` в настройках meson.
- `libraw`, `libtiff`, `libjpeg`, `libpng`, `libheif` для RAW, TIFF, JPEG, PNG и HEIF images import and export. The libraries are detected at compilation-time.
- `FFMS2` для нативной поддержки видео как последовательностей изображений. Это также позволяет извлекать кадры из многих видов видео для других целей, помимо астрономии. В версиях < 2.20 есть досадная ошибка, поэтому рекомендуется установить последнюю версию.
- `ffmpeg` (или `libav`), предоставляющая `libavformat`, `libavutil` (>= 55.20), `libavcodec`, `libswscale` и `libswresample` для экспорта последовательностей mp4.
• **gnuplot** для создания графиков фотометрии (во время компиляции не требуется).

• **wcslib** для управления мировой системой координат, аннотаций и фотометрической калибровки цвета.

• **libconfig** (поддержка структурированных конфигурационных файлов), используется для чтения файла конфигурации с версий до 1.0. Сейчас используется только для получения старых настроек.

• **libjson-glib** для проверки обновлений (бесполезно, если вы создаете нерелизную версию).

• **Exiv2** для управления метаданными изображения.

• **libcurl ИЛИ glib-networking** с его HTTP-серверной частью для удалённых операций, таких как проверка обновлений, запросы на астрометрию и фотометрию.

### Зависимости для компиляции

Для установки из исходных кодов, вам необходимо установить базовые пакеты разработки:

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

Компиляторы gcc и g++ из этого списка могут быть заменены на clang и clang++ (мы используем их для разработки), и, возможно, другими.

Пакеты autotools (autoconf, automake, возможно, некоторые другие) могут быть заменены на meson.

### 1.4.3 Типичный процесс сборки

Siril может быть скомпилирован с помощью autotools или meson.

**Meson**

Рекомендуемый способ это использовать meson и ninja:

```
meson setup _build --buildtype release

cd _build
ninja
ninja install
```

To disable some dependencies or features, use meson options `-Dfeature=false` or `-Denable-feature=yes` depending on the case.

Table below lists all configurable options.
<table>
<thead>
<tr>
<th>Option</th>
<th>Type</th>
<th>Value</th>
<th>Choices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>relocatable-bundle</td>
<td>combo</td>
<td>platform-default</td>
<td>['yes', 'no', 'platform-default']</td>
<td>build with resources considered bundled under the same prefix</td>
</tr>
<tr>
<td>openmp</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with OpenMP support</td>
</tr>
<tr>
<td>json_glib</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with json glib support</td>
</tr>
<tr>
<td>exiv2</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with exiv2 support</td>
</tr>
<tr>
<td>libraw</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with LibRaw support</td>
</tr>
<tr>
<td>libtiff</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with TIFF support</td>
</tr>
<tr>
<td>libjpeg</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with JPEG support</td>
</tr>
<tr>
<td>libpng</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with PNG support</td>
</tr>
<tr>
<td>libheif</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with HEIF support</td>
</tr>
<tr>
<td>ffms2</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with FFMS2 support</td>
</tr>
<tr>
<td>ffmpeg</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with FFmpeg support</td>
</tr>
<tr>
<td>enable-libcurl</td>
<td>combo</td>
<td>platform-default</td>
<td>['yes', 'no', 'platform-default']</td>
<td>Use libcurl instead of GIO</td>
</tr>
<tr>
<td>libconfig</td>
<td>logical</td>
<td>false</td>
<td>Н/Д</td>
<td>build with libconfig support</td>
</tr>
<tr>
<td>criterion</td>
<td>logical</td>
<td>false</td>
<td>Н/Д</td>
<td>build with criterion support</td>
</tr>
<tr>
<td>wcslib</td>
<td>logical</td>
<td>true</td>
<td>Н/Д</td>
<td>build with WCSLIB support</td>
</tr>
</tbody>
</table>

**Autotools**

Компиляция с помощью autotools хорошо известна в мире Unix. После загрузки исходных кодов и установки всего необходимого, обычный способ компиляции следующий:

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

возможно с правами суперпользователя для последней строки.

Возможно, вы захотите передать компилятору определенные параметры, например, такие, если бы вы хотели оптимизацию и установку в `/opt` вместо `/usr/local`, используемого по умолчанию:

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

Для запуска Siril имя команды `siril` или `siril-cli`.

**1.4.4 Установка в Debian и подобных системах**

Возможно, вы захотите создать пакет .deb вместо использования неупакованной версии, в этом случае обратитесь к этой справке. В частности, для установки зависимостей вы можете использовать команду:

```
apt build-dep siril
```

В противном случае вот список пакетов для текущей версии:

- Пакеты, необходимые для компиляции системы:

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

1.4. Установка из исходных кодов
Пакеты с обязательными зависимостями:

```
libfftw3-dev libgsl-dev libcfitsio-dev libgtk-3-dev libopencv-dev
libexiv2-dev
```

Пакеты с необязательными зависимостями:

```
wcslib-dev libcurl4-gnutls-dev libpng-dev libjpeg-dev libtiff5-dev
libraw-dev gnome-icon-theme libavformat-dev libavutil-dev libavcodec-dev
libswscale-dev libswresample-dev
```

для поддержки видео (AVI и прочие):

```
libffms2-dev
```

### 1.4.5 Установка в Arch Linux

В AUR доступны два пакета: `siril` и `siril-git`. Скачайте `PKGBUILD` или репозиторий, установите зависимости, запустите `makepkg` для создания пакета и `pacman -U` для его установки.

Зависимости (обязательные и несколько необязательных):

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

### 1.4.6 Неудачи сборки

Каждый коммит в git Siril автоматически собирается в стандартной среде сборки для Linux, Windows и MacOS с использованием инфраструктуры gitlab CI. Это значит, что мы с высокой долей вероятности уверены в том, что компиляция основной ветки и отмеченных выпусков будет успешной при условии корректно настроенной среды сборки и наличии всех необходимых зависимостей.

Если вам не удаётся скомпилировать программу, это, скорее всего, свидетельствует о проблеме с вашей средой для сборки или неверно установленных зависимостях. Во многих дистрибутивах требуется отдельная установка пакетов разработки, которые содержат необходимые заголовочные файлы. Проверьте сообщение CI из того коммита, который вы пытаетесь скомпилировать. В маловероятном случае, если действительно будет показан сбой сборки, будьте уверены, команда работает над его устранением. В противном случае, если конвейер CI показывает зеленые галочки, вам нужно будет просмотреть и устранить любые проблемы с вашей собственной средой сборки.

Если вы все еще считаете, что обнаружили проблему сборки, не отмеченную конвейером CI — например, если вы осуществили сборку на другой платформе, такой как BSD, которую разработчики не используют регулярно, — тогда не стесняйтесь говорить о проблеме на `gitlab`.

Обратите внимание, что о проблеме следует сообщать только в отношении основной ветки или отмеченных выпусков. Если вы тестируете новые функции в запросах на слияние, пожалуйста, оставьте отзыв в комментариях к соответствующему запросу.
Глава 2

Графический интерфейс пользователя

Графический интерфейс пользователя позволяет обрабатывать ваши изображения вручную, равно как использовать сценарии или вводя команды. Чтобы узнать, как использовать Siril в автономном режиме (headless mode), обратитесь к этому разделу.

Графический интерфейс Siril написан с использованием GTK, свободной, кроссплатформенной библиотеки для создания графических пользовательских интерфейсов. В настоящее время используется GTK3.

В последующих разделах будут описаны основное окно и полезные меню.

2.1 Основной интерфейс

После запуска Siril, открывается основной интерфейс.

Примечание: Нажмите на любом элементе изображения основного интерфейса ниже, чтобы узнать о его функциях.
Область изображения

В этой области отображается загруженное в данный момент изображение. Нажмите на Красный, Зелёный или Синий, для переключения между разными слоями (только для цветных изображений, для монохромных доступна отдельная вкладка Ч/Б).

При нажатии правой клавишей на изображении, появляется контекстное меню:
Совет: Когда не загружено ни одного изображения, двойное нажатие в области изображения вызывает диалог открытия файлов.

План: Объяснить или дать ссылку (когда будет создана) на другие элементы

Открыть

Нажмите на эти иконки (слева направо) чтобы:

- открыть файл
- открыть недавний файл
- сменить домашний каталог

Вернуться к иллюстрации
Укладка на лету

Нажмите эту кнопку, чтобы начать сессию Укладка на лету.

Вернуться к иллюстрации

Отменить/Повторить

Используйте эти кнопки для отмены/повтора последних действий. Возможно только в том случае, если последнее действие было выполнено через пользовательский интерфейс, а не путём ввода команды.

Вернуться к иллюстрации

Обработка изображений

Нажмите на эту кнопку, чтобы вызвать меню Обработка.

Вернуться к иллюстрации

Сценарии

Нажмите на эту кнопку, чтобы показать список доступных сценариев и запустить сценарий.

Вернуться к иллюстрации

Информационная строка

В этой панели отображается информация о версии Siril и текущем рабочем каталоге.

• Справа приведена информация о доступном объёме оперативной памяти и дискового пространства.
• Вы можете так же менять количества доступных Siril потоков, используя значки +/–.

Вернуться к иллюстрации

Сохранить

Эти кнопки используются для сохранения результатов:

• сохранить (перезаписать) текущее изображение.
• сохранить с другим именем и/или расширением.

Выпадающий список в правом нижнем углу позволяет вам выбрать тип записанного изображения. Он автоматически добавляет расширение к имени файла. Однако, оставаясь в режиме Поддерживаемые форматы, можно вручную добавить любое расширение, поддерживаемое Siril, и программа сохранит изображение в правильном формате.
• сделать снимок текущего вида изображения (включая растягивание, если такое применяно). Возможны два варианта. Либо снимок сохраняется в буфере обмена, либо непосредственно копируется на диск в рабочем каталоге.
• изменить разрядность текущего изображения. Доступны 16 и 32 битные режимы.

Вернуться к иллюстрации
Рис. 1: Диалог сохранения файлов.
Главное меню

Открывает главное меню (оно же burger menu). Позволяет получить доступ к настройкам, решению изображений и многому другому.

Вкладки

Выбор одной из вкладок. Вы можете переключаться между вкладками, используя клавиши F1 – F7. Больше подробной можно найти:

- Вкладка Конвертация
- Вкладка Последовательность
- Вкладка Калиборовка
- Вкладка Регистрация
- Вкладка График
- Вкладка Укладка

Командная строка

Наберите команду и нажмите Ввод.

- Вы можете нажать кнопку в конце командной строки, чтобы получить помощь по использованию команды.
- Чтобы прервать текущий процесс, нажмите на кнопку Стоп.

Расширить

Нажмите на эту панель, чтобы показать/скрыть всю вкладку.
Ползунки

Используйте верхний и нижний ползунки, чтобы настроить, соответственно, точки белого и чёрного при предварительном просмотре изображения (в линейном режиме).

Совет: Нажатие на название загруженного в данный момент изображения или последовательности позволяет скопировать это название в буфер обмена (полезно для вставки в команду).

Предварительный просмотр

Выберите режим предварительного просмотра загруженного изображения из следующих:

- Линейный
- Логарифмический
- Квадрат. корень
- Квадратичный
- Asinh
- Autostretch (tick the High Definition box to use a deeper 20-bit LUT instead of the default 16-bit one)
- Гистограмма

В режиме Авторастягивания со цветными изображениями переключатель справа включает/отключает привязку каналов. При разъединении, 3 слоя растягиваются независимо друг от друга, чтобы получить более сбалансированное изображение.

Предупреждение: Это лишь предварительный просмотр изображения, а не реальные данные (за исключением Линейного режима). Не забудьте растянуть ваши изображения перед сохранением.

Специализированные режимы просмотра

Используйте эти переключатели, для предварительного просмотра изображений:

- в виде негатива
- в ложных цветах

2.1. Основной интерфейс 21
Инструменты астрометрии

Используйте эти переключатели, чтобы показать:

- астрономические аннотации
- небесную сетку

Предупреждение: Чтобы эти кнопки были активны, загруженное изображение должно иметь астрономическое решение.

Быстрая фотометрия

Используйте эту кнопку, чтобы переключиться в режим быстрой фотометрии.

Масштабирование

Используйте эти кнопки чтобы:

- Отдальять изображение
- Приблизить изображение
- Подогнать изображение под размеры окна
- Показать изображение в полный размер

Совет: Ctrl+left clic will allow to navigate into the picture

Совет: Ctrl+прокручивание колеса мыши приближают/отдаляют изображение, а Ctrl + 0 / 1 подгоняют масштаб к размеру окна/100%, соответственно.

Геометрические преобразования

Используйте эти кнопки чтобы:

- Повернуть влево
- Повернуть вправо
- Отразить по горизонтали
- Отразить по вертикали

Вернуться к иллюстрации
Выбор кадров

Нажмите на эту кнопку, чтобы открыть диалог выбора кадров.

Вернуться к иллюстрации

2.2 Главное меню

2.2.1 Первая страница

Данные об изображении

Доступ к следующей странице меню.

Настройки

Открывает меню настроек.

Руководство Siril

Открывает сайт с документацией.

Проверка обновлений

Проверяет наличие новой версии.

Скачать сценарии

Открывает страницу, где можно скачать сценарии, в дополнение к уже имеющимся после установки Siril.

Клавиатурные сокращения

Открывает окно со всеми доступными Клавиатурные комбинации.
О программе

Открывает диалог с информацией о версии и благодарности.

2.2.2 Вторая страница

Астрономическое решение

Открывает диалог поиска астрономического решения.

Статистика

Открывает диалог Статистика.

Оценка шума

Производит оценку шума загруженного изображения. Результаты будут выведены в журнале.

Аберрации

Открывает диалог aberrаций.

Информация

Открывает диалог с данными об изображении.
Заголовок FITS

Отображает заголовок FITS текущего изображения.

Динамическая PSF

Открывает диалог Динамической PSF.
2.3 Клавиатурные комбинации

Siril использует несколько клавиатурных сочетаний для доступа к инструментам обработки или для манипулирования приложением и/или изображениями. Эти сочетания клавиш приведены в диалоге Клавиатурные комбинации, доступном через главное меню.

![Клавиатурные комбинации](image)
2.4 Окно с данными об изображении

Информация относится к изображению, извлеченной из файла, если доступна, иначе с учетом настроек. Модифицированные данные можно сохранить в файле FITS.

Поля:
- **Focal length (mm):** 252.000
- **Pixel size (micron):** 3.76, 3.76
- **Dinning:** 1x1
- **Computed sampling (arcsec/pixel):** 3.0776

Опции:
- Clear
- Save as default values

Закрыть окно.
В этом окне содержится информация о выборке открытого изображения. Выборка, называемая также разрешение или масштаб, представляет собой угловой размер пикселя при наблюдении через этот прибор. Она зависит от двух факторов: фокусного расстояния прибора и размера пикселя в сенсоре, который, в свою очередь, зависит от режима биннинга.

Заголовок FITS содержит эту информацию, если она была передана программному обеспечению для захвата изображений. В таком случае эти значения отображаются в информационном окне. Если же метаданные изображения не содержат этих значений, поскольку они не были известны программе захвата или, проще, потому что формат файла их не поддерживает, данное диалоговое окно всё равно будет доступно и заполнено значениями по умолчанию. Они могут быть изменены и использованы для различных операций Siril, требующих информацию о масштабе, например, отображение FWHM в угловых секундах вместо пикселей.

Значения по умолчанию — биннинг отсутствует (1×1), а фокусное расстояние и размер пикселя сохранены в настройках. Значения, сохранённые в настройках, можно задать из этого диалогового окна, установив флажок Сохранить как значения по умолчанию, прежде чем нажать на кнопку Закрыть. Их также можно задать, выполнив астрометрическое решение изображения, также называемое решение пластин (plate solving), если в настройках включена опция обновления значений по умолчанию когда найдено астрономическое решение.

Значения, отображаемые в этом окне, будут сохранены в текущем загруженном изображении, и, если это изображение сохранено как FITS, они будут сохранены в заголовке FITS.

Управление биннингом может осуществляться в двух режимах, в зависимости от программного обеспечения для захвата изображений: задаётся реальный размер пикселя, но его необходимо умножить на коэффициент биннинга (когда установлен флажок Реальный размер пикселя), или задаётся размер пикселя, уже умноженный на коэффициент биннинга (когда флажок снят).
Глава 3

Настройки

Настройки это те предпочтения, которые сохраняются для каждого сеанса работы с Siril и определяющие предпочитаемый вами выбор для многих инструментов.

Начиная с версии 1.2.0, настройки доступны как через пользовательский интерфейс, так и программно, с помощью команд set/get.

По умолчанию файл с настройками расположен:

- `~/.config/siril/config.ini` (Linux)
- `%LOCALAPPDATA%sirilconfig.ini` (Windows)
- `~/Library/Application Support/org.free-astro.Siril/siril/config.ini` (MacOS)

Если вы хотите иметь несколько файлов с конфигурацией, вы можете выбрать какой использовать, открыв терминал и набрав:

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

**Предупреждение**: Чтобы использовать `siril` как показано строкой выше, необходимо, чтобы Siril был в системной переменной path. В противном случае необходимо указать полный путь к исполняемому файлу Siril.
3.1 Настройки (графический интерфейс)

Настройки доступны через главное меню или через клавиатурное сочетание Ctrl + P. В настройках 10 страниц, каждая из которых освещает определённую группу функций. Настройки позволяют более или менее продвинутым пользователям оптимизировать Siril для наилучшего удовлетворения их потребностей. Некоторые настройки могут негативно воздействовать на работу Siril, поэтому желательно изменять настройки только тогда, когда вы знаете, что делаете. В нижней части окна настроек расположены три кнопки: Сброс сбрасывает все настройки к значениям по умолчанию, Отмена отменяет текущие изменения, а Применить закрывает диалоговое окно и сохраняет настройки.

3.1.1 Дебайеризация FITS/SER

Вкладка Дебайеризация FITS/SER позволяет пользователю определить настройки дебайеризации файлов FITS, SIR или TIFF. Следовательно, эта вкладка полезна только пользователям со цветной камерой. Желательно оставить настройки по умолчанию, так как Siril автоматически определяет правильные настройки. Однако в случае файла TIFF, не являющегося AstroTIFF, или файла, в котором отсутствуют все необходимые ключевые слова, может потребоваться изменить настройки вручную. В этом случае вам необходимо снять флажок с кнопки По возможности брать информацию о шаблоне.

Рис. 1: Страница 1 диалога настроек
Байера из заголовка файла. Это действие сделает доступными несколько настроек, которые пользователь может изменить.

- **Шаблон Байера:** Это выпадающее меню позволяет выбрать шаблон Байера, используемый камерой. Как правило, это указано в информации от производителя. Однако будьте осторожны, это поле тесно связано с параметром Дебайеризация файлов FITS сверху вниз, если не найдены явные ключевые слова, и результаты будут отличаться независимо от того, установлен этот флажок или нет. Более подробное объяснение последнего варианта можно найти здесь.

- **Смещение по X:** В редких случаях файлы записываются со сдвигом массива Байера. Мы можем определить смещение в 1 по оси X и смещение в 1 по оси Y. Указанное здесь значение определяет смещение по X.

- **Смещение по Y:** смещение массива Байера по Y.

Изменение этих настроек каждый раз будет включать в себя иную дебайеризацию, поэтому настоятельно рекомендуется оставить настройки в значениях по умолчанию, если только вы действительно не уверены в том, что делаете.

Следующая настройка, которая оказывает меньшее влияние на конечный результат, является алгоритм дебайеризации, доступный в пункте Интерполяция. Возможны следующие варианты:

- **Быстрая дебайеризация** это самый быстрый алгоритм, доступный в Siril. Однако другие алгоритмы, перечисленные ниже, часто дают гораздо лучший результат.

- **VNG4** (Threshold-Based Variable Number of Gradients), алгоритм работает в области 5х5 пикселей вокруг каждого исходного пикселя. Это очень хороший алгоритм для ровных областей изображения (например, фон неба), но создает артефакты в высококонтрастных областях (например, звезды).

- **AHD** (Adaptive Homogeneity-Directed), это другой хорошо известный алгоритм дебайеризации. Однако обычно он демонстрирует артефакты фона и плохие формы звёзд.

- **AMaZE** (Aliasing Minimization and Zipper Elimination), это алгоритм, который даёт хорошие результаты, особенно при съёмке с низким уровнем шума.

- **DCB** (Double Corrected Bilinear), более современный алгоритм, демонстрирующий некоторые артефакты фона, подобно AHD.

- **HPHD** (Heterogeneity-Projection Hard-Decision), старый алгоритм, демонстрирующий хорошие результаты, но достаточно медленный.

- **IGV and LMMSE** очень хороши при работе с сильно зашумлёнными изображениями. Однако IGV имеет тенденцию к потере некоторой цветовой информации, а LMMSE — один из алгоритмов дебайеризации требующих значительного количества вычислений и который нуждается в большом количестве памяти.

- **RCD** (Ratio Corrected Demosaicing) предназначен для сглаживания ошибок в коррекции цвета, которые имеют место у многих методов интерполяции. Этот алгоритм хорошо себя ведёт на круглых границах, например, звёздах, и поэтому является алгоритмом по умолчанию в Siril.

Для сенсоров X-Trans, независимо от указанного в настройках, используется специальный алгоритм Markesteijn. Для этого алгоритма возможно указать желаемое качество с помощью опции Качество дебайеризации X-Trans. Она определяет количество проходов алгоритма дебайеризации X-Trans Markesteijn. Значение по умолчанию — 1, немного лучшим, но более медленным будет 3.

**Предупреждение:** Для дебайеризации на лету файлов *SER всегда используется алгоритм RCD, независимо от выбора, сделанного в выпадающем меню. Это позволяет Siril быть более эффективным во времени выполнения и обеспечивает хорошее качество.
3.1.2 Параметры FITS

На странице Параметры FITS сгруппированы все настройки, связанные с нативным форматом файлов, используемым Siril.

- Расширение: по умолчанию значение установлено в .fit. Однако многие программы захвата используют расширение .fits. В таком случае необходимо обновить это значение. Все файлы, создаваемые Siril, будут использовать расширение, указанное здесь. Более того, могут быть загружены только те последовательности, которые имеют расширение, указанное в настройках. Таким образом нельзя одновременно открыть последовательности .fits и .fit, не обновив это значение. Поддерживаются следующие расширения:
  - .fit
  - .fits
  - .fts

Ко всем им можно прибавить расширение .fz, если файл сжатый.

Командная строка Siril
Устанавливает расширение, используемое и понимаемое последовательностями.

Аргумент **extension** может быть "fit", "fts" или "fits"

- **Тип по умолчанию:** По умолчанию Siril работает с 32 битными числами с плавающей точкой в диапазоне \([0, 1]\). Это лучший способ иметь высокую точность. Однако из соображений места, занимаемого файлами на жёстком диске, пользователь может принять решение работать с 16 битными числами без знака (в диапазона \([0, 65535]\)). Будьте осторожны, поскольку в этом случае возможна потеря большого количества информации при укладке.

- **Кубы FITS могут содержать изображения разного размера:** Это может быть полезно при использовании научных файлов FITS, которые не созданы в Siril и содержат множественные изображения разного размера. В противном случае такие файлы рассматривались бы как неверные файлы Siril FITSEQ. Хорошим примером использования это опция являются изображения космического телескопа "Джеймс Уэбб".

- **Включить совместимость с Aladin (CTYPE3 = 'RGB '):** Aladin рассматривает 3D-куб FITS как изображение RGB (Красный, Синий и Зелёный компоненты), если в заголовке указано ключевое слово FITS CTYPE 3 = 'RGB'. В этом случае поддерживаются любые значения BITPIX. Без установки ключевого слова FITS CTYPE3 = 'RGB' только куб FITS с 3 кадрами одинакового размера и с BITPIX=8 будет автоматически определен как RGB FITS.

**Предупреждение:** Эта опция может конфликтовать с функцией астрометрии и должна быть включена если она действительно необходима.

- **Обновить размер пикселей при биннинге:** Используется для вычисления масштаба изображения. Размер пикселя может быть задан двумя способами: указывается реальный размер пикселя, который умножается на коэффициент биннинга (если флажок установлен), или указывается размер пикселя уже умноженный на коэффициент биннинга (если флажок снят). Данная настройка зависит от программы захвата, использованной для создания файлов FITS.

- **Сжатие FITS:** Сжатие может иметь смысл в тех случаях, когда пространство на жёстком диске является важным аспектом. Больше информации доступно в разделе, посвящённом формату FITS.

При сжатии к файлу добавляется расширение .fz. Siril имеет возможность открывать последовательность с расширением fz без необходимости изменения настроек.

**Командная строка Siril**

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

Определяет, сжимать изображения или нет.

0 означает, что сжатие выключено, а 1 включает сжатие.
Если сжатие включено, его тип должен быть явно указан опцией -type= ("rice", "gzip1", "gzip2").
Связанное со сжатием значение квантования должно быть в диапазоне \([0, 256]\). Например, "set compress 1 -type=rice 16" устанавливает тип сжатия (rice) со значением квантования 16

### 3.1.3 Астрометрия

На данной вкладке располагаются настройки, связанные с астрометрией, которая является важным инструментом Siril. После получения астрономическое решение изображения, т.е. астрометрия была успешной, возможно показать названия известных объектов. В частности тех, которые указаны в крупных астрономических каталогах. Раздел Аннотации позволяет указать, какие каталоги могут быть использованы для показа названий объектов. В настоящее время этих каталогов 6 и они могут быть исключены из выделения, чтобы Siril их игнорировал:

- Каталог Мессье (M)
- Новый общий каталог (NGC)
- Индекс-каталог (IC)
• Кatalog ярких туманностей Линда (LdN)
• Кatalog Шарплесса (Sh2)
• Кatalog ярчайших звёзд

В дополнение к этому списку, имеются два каталога, которые наполняются пользователем. Первый каталог — объекты глубокого космоса и второй — объекты Солнечной системы. Эти каталоги лучше описаны в разделе документации, посвящённому аннотациям.

После нажатия на кнопку Показать имена небесных тел (только если изображение имеет астрономическое решение), на изображении отображаются аннотации. Так же возможно отобразить небесную сетку, нажав на кнопку . Последняя, по умолчанию, добавляет компас в центре изображения.

Раздел Мировая система координат позволяет выбрать

• Formalism 1: In the PC i,j formalism, the matrix elements $m_{ij}$ (linear transformation matrix) are encoded in PC i,j (floating-valued) header cards, and si as CDELT i. The i and j indices are used without leading zeroes, e.g. PC 1,1 and CDELT 1. The default values for PC i,j are 1.0 for $i = j$ and 0.0 otherwise. The PC i,j matrix must not be singular; it must have an inverse. Furthermore, all CDELT i must be non-zero.

• Formalism 2: The CD i,j (floating-valued) keywords encode the product $s_i m_{ij}$. The i and j indices are used without leading zeroes, e.g. CD 1,1. The CD i,j matrix must not be singular; it must have an inverse. CDELT i and CROTA i are allowed to coexist with CD i,j as an aid to old FITS interpreters, but are to be ignored by new readers.

В разделе диалогового окна Локальные каталоги находятся локальные каталоги, используемые для поиска астрономического решения изображений. Подробности описаны в разделе Аннотации.

Опция в разделе Настройки решателя определяет, сохранять ли после нахождения астрономического решения фокусное расстояние и входящий размер пикселя как значения по умолчанию для изображений, не имеющих соответствующих метаданных.

Последний раздел посвящён решателю пластин solve-field из набора astrometry.net.

• Степень полиномиальной коррекции: astrometry.net может использовать полиномиальную коррекцию (SIP) для работы с оптическими искажениями. Здесь указывается степень полиномиальной модели. Значение 0 отключает её.

• Допуск масштаба: процент, определяющий допустимую нижнюю и верхнюю границы масштаба. Данный масштаб умножается или делится на $1 +$ это значение / 100.

• Радиус цели: допустимый радиус (в градусах) вокруг координат цели для решения. Не используется для решения вслепую (без указания цели).

• Не удалять таблицы .xyls: список звёзд передаётся команде solve-field в виде таблицы FITS. Установите этот флажок, чтобы сохранить этот файл в рабочей директории.

• Не удалять файлы .wcs: результаты команды solve-field хранятся в заголовке FITS файла, оканчивающимся .wcs. Установите флажок, чтобы не удалять этот файл.

• Максимальное кол-во секунд для попытки решения: допустимое время поиска решения для каждого файла каталога. Может использоваться как общее время для решения только в том случае, если команда solve-field сконфигурирована таким образом в файле настроек.

• Location of local astrometry.net solver: In order to use Astrometry.net locally in Siril, it can be necessary to tell to Siril the path where it is located. On UNIX-based systems, it is generally in the PATH variable and not necessary. For Windows, if you did not modify the default installation directory,
• Показать вывод solve-field: отображать вывод решателя в журнале Siril. В противном случае будет отображён только результат работы решателя.

3.1.4 Предобработка

Во вкладке "Предобработка" располагаются элементы, связанные с шагами, выполняющимися до завершения укладки. В частности, здесь можно управлять библиотеками кадров шума считывания, темновых кадров и кадров плоского поля, именем файла, полученного в результате укладки или специфичными коррекциями для камер, использующих сенсор X-Trans.

• Библиотеки темновых кадров, кадров шума считывания и плоского поля: В этом разделе можно загрузить мастер-кадр шума считывания, темновой мастер-кадр и мастер-кадр плоского поля, которые будут использованы по умолчанию в ходе предобработки, если справа от текстового поля установлен флажок Использовать по умолчанию. Каждый путь так же сохраняется в зарезервированном ключевом слове $defbias, $defdark и $defflat (один знак $), которые может использоваться при сохранении результатов укладки. Что касается шума считывания, можно использовать не только путь к файлу. Действительно, в команде Siril мы призываем пользователей использовать синтетические кадры шума считывания, как это описано в этом руководстве. Тогда возможны несколько значений, если первым символом является знак =. Возможно использовать фиксированное целое значение, например =500 или умножение с применением ключевого слова $OFFSET (один знак $), при условии, что последнее зарегистрировано в заголовке файла FITS, например 10*$OFFSET. Больше подробностей приведено в соответствующем руководстве.

• Stacking default: Here we define the default name that we want to give to the stacking results. It is possible to use any value given in the FITS header as a keyword and surround it with $ tokens.
Рис. 5: Страница 4 диалога настроек
If the keyword does not exist the variable will be used, otherwise it is its value. Another reserved keyword that can be used is $\texttt{seqname}$. It contains the name of the loaded sequence. For example, the following default name, $\texttt{seqname}\texttt{stacked}_\texttt{LIVETIME:s}$ with a sequence name $r_{pp\_light}$ and the following 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) 
... 
```

will output $r_{pp\_light}\texttt{stacked\_3900s.fit}$.

- **Fix Xtrans files**: This setting field is very specific and only concerns possessors of certain X-Trans sensors. Indeed, some images from these cameras show a large square in the center of the darks and bias images due to the position of the autofocus (AF). Siril has an algorithm to eliminate it for the following cameras:
  - Fujifilm X-T1
  - Fujifilm X-T2
  - Fujifilm X-T20
  - Fujifilm X-Pro2
  - Fujifilm X-E3
  - Fujifilm X-H1

Рис. 6: X-Trans artifact fixed by the algorithm of Siril

In the unlikely event that your camera contains this artifact and is not supported, then it is possible to define the correction to be applied here. The best thing to do is to contact the dev team in order to have the values to enter that would correspond to your camera.
3.1.5 Фотометрия

Photometry, which is the study of light, is another feature very present in Siril. This section of the preferences allows you to define the settings associated with this tool.

The basic principle of aperture photometry is to sum-up the observed flux in a given radius from the center of an object, then subtract the total contribution of the sky background in the same region (calculated in the ring between the inner and outer radius), leaving only the flux of the object to calculate an instrumental magnitude. This is described in more detail in the Фотометрия section of this documentation.

- It is then possible to modify the inner radius and the outer radius to define a size that optimizes the calculated sky value, trying to avoid the stars inside the ring. Outer must always be greater than inner. By default, the flux aperture radius is set as twice the PSF's FWHM, however it is possible to disable this feature and define a fixed value manually.

- Pixel range value allows users to set a limit for which the pixel is considered bad for photometry. Indeed, doing photometry on saturated data will never give good results, but even getting close to high values may not be suitable because it may be in the non-linear regime of sensors. A default value of 50000 ADU is set to avoid this region, but it may vary from sensor to sensor. Negative values are also allowed because noise can average around a positive value but still provide a few pixels with negative values.
Рис. 8: Circle of the aperture photometry
• Finally, if known, it is highly recommended to put the value of the A/D converter gain in electrons per ADU: it is used in the uncertainties computations, if not already provided in the headers of the processed images.

3.1.6 Инструменты анализа

![Preferences dialog](image)

Рис. 9: Page 6 of preferences dialog

So far, only one image analysis tool requires adjustment parameters. It is the aberration inspector tool. In this tab you can adjust:

• The **panel size**, in pixels, which defines the size of the image that will be placed in a panel. The larger the value, the larger the size of the image in a panel. A value that is too high may prevent from seeing the defects of the stars.

• The **window size**, also in pixels, which defines the size of the dialog. It is usually a good idea to increase this value when using a 4K screen.
Рис. 10: Aberration inspector window
3.1.7 Интерфейс

In this tab are listed all the adjustments related to the user interface. These are not settings that have an impact on the processes, but on the feel and look and needs of the user.

- By default, the **language** of Siril is defined according to the system language. However it is possible to change the language and define it to your needs, as long as it exists. However, keep in mind that Siril is developed in English.

- Two themes are available:
  - The dark theme (default theme)
  - The light theme

  Changing the theme requires a restart of the application to be fully operational.

- It is possible to adjust the **font scale** for users with a 4K Ultra-HD screen, or to use **symbolic icons** for some icons. Theses settings also require a restart of the application.

- By default Siril remembers the size and the position of the application window each time you close it. By checking on the **Remember window size and position** button you can disable this behavior.
• The thumbnails of the images are usually visible in the open dialog boxes. The preferences allow you to not display them if the computer has limited performance and the user does not see the need. You can also change the size of the thumbnail display with the drop-down list.

• **Default screen transfer function** is the setting that allows images to be displayed according to the user's preference. By default, this is set to linear. As this really represents what the image is, it is recommended that beginners leave this setting at default. It is easy to forget that you are in auto-adjusted viewing and not understand why the saved images are not as they appear on the screen. However you can always adjust the visualization in the main window.

• **Default display mode**: According to the same principle, the histograms can be displayed in two modes. Either the linear or the logarithmic mode. The latter can be very useful with the Generalized Hyperbolic Stretch tool. You can however change the mode in each window with a histogram. In the preferences, this is a matter of setting the default behavior.

3.1.8 Сценарии

![preferences dialog](image)

Рис. 12: Page 8 of preferences dialog

• The Scripts tab essentially contains the locations where Siril should look for scripts. Indeed, by default and depending on the OS used, the scripts are installed in a specific place:
– /usr/local/share/siril/scripts or /usr/share/siril/scripts on GNU/Linux.
– C:\Program Files\Siril\scripts on Windows.
– /Applications/Siril.app/Contents/Resources/share/siril/scripts on MacOS, if the application has been installed in the Applications folder.

Предупреждение: On macOS, as the application is signed and notarized, it is impossible to modify the scripts inside the bundle. Otherwise, the application will not start. So you have to define another path pointing to a folder where you have write permissions.

- The **Script Storage Directories** field allows you to define custom folder paths to place scripts that you have created and/or modified. Clicking on the button just below will rescan the folders and update the list of scripts in the dedicated menu.

- The **Warning Dialogs** section proposes to disable:
  1. The warning text that is displayed before a script is executed.
  2. The check of the keyword *requires* which must be at the very beginning of the script in order to check if the script is compatible with the version of Siril. We recommend to not uncheck this option.

### 3.1.9 Производительность

Astronomical image processing software, such as Siril, uses a lot of resources and usually requires quite powerful computers. It is not impossible, when the computer is very busy, that it freezes completely. It is not at all recommended to do anything else on the computer during the processing, especially Internet browsing, because browsers are very greedy for RAM. However, it is possible to manage the maximum percentage of RAM that Siril can use.

- **Ratio of available**: Siril will limit itself to a ratio of the amount of free physical memory and will decrease the size of work tasks if needed. A value above 1 means that some memory paged on a configured storage will be used and that the overall process will be slower and the system will likely be unresponsive during some operations. If you don’t have paged memory configured on some storage, a value of 1 or above will likely result in a crash of siril or of the operating system.

- **Fixed amount (GB)**: Siril will limit itself to a fixed amount of memory and will decrease the size of work tasks if needed. Configuring a memory amount larger than what is available on your system may result in a crash of siril or of the operating system.

- The **Default bit depth for HD AutoStretch** option sets default bit depth for the HD AutoStretch display mode. Higher bit depths require exponentially more memory for the LUT and take longer to recalculate it, but do a much better job of smoothing quantization artefacts in displaying images with very narrow histogram peaks. The default bit depth will apply from the next viewer mode change, and can be applied now using the button to the right. Click on the **Apply bit depth** button to set the selected HD AutoStretch bit depth now.

- **Multithreaded FFTW**: this toggle button sets FFTW to use multiple threads. This can be faster (though performance does not increase linearly with the number of processors, due to synchronization overhead), but FFTW’s planning stage takes longer for multi-threaded systems so particularly the first FFT for a given image size may be considerably slower using multiple threads.

- **FFTW planning strategy**: this combobox sets the FFTW planning strategy. FFTW has multiple algorithms for calculating a FFT and will plan a given FFT to optimize speed. It saves the results of these plans for later reuse in a cache file called "Wisdom", so some extra time spent up-front planning can reap rewards if you calculate a lot of FFTs of the same size. Note that wisdom is specific to a
Рис. 13: Page 9 of preferences dialog
given machine: it should not be shared between machines and should be deleted and regenerated from
scratch following a memory or processor upgrade or a major change of software environment (major
OS changes, Siril major version changes). In order of speed, Estimate is fastest: this strategy does not
actually do any measurement but does planning based on a set of heuristics. Measure is next fastest:
this method actually compares the speed of different internal FFTW methods of calculating the FFT
and picks the fastest. As a result, the planning step takes longer. Patient considers even more possible
plans, and Exhaustive considers even more. If you always process images of a specific size then the more
expensive planning strategies may be worthwhile because of Wisdom, but if you work with images of
lots of different sizes then a cheaper planning algorithm may be more suitable.

- **FFTW planning time limit**: this time limit halts FFTW planning after the specified time limit.
  This will override the planning strategy. Note that the time limit is not strictly enforced: FFTW will
  finish any non-interruptible calculation it is performing at the time the limit is reached, and if set to
  zero FFTW will always as a minimum carry out Estimate planning.

### 3.1.10 Разное

The last tab contains everything that does not fit in the other themes.

- **Using the *Undo*/*Redo* buttons requires disk space. Lots of space in some cases. The folder containing
  the swap files (which are the files necessary for the proper functioning of the undo/redo pair) can
  be defined in the **Swap Storage Directory** section. The disk space is listed to the right of the file
  chooser. We advise **to not change** the default settings unless you have a good reason to do so. As the
  choice of a good folder is critical, it is possible to return to the default folder by clicking on *Restore to
  Default*.

- **The **Warnings Dialogs** allows to disable some warning popups that are here to help beginners.**

- **Introduction Tips**: At the very first start of Siril, it is possible to see a little animation showing
  what’s new in the application. This animation can be replayed by clicking on *Play introduction*.

- **StarNet Executable Location**: In order to use StarNet in Siril, it is necessary to tell to Siril the
  path where the StarNet executable is located. For old StarNet++ v1 installations that use separate
  executables to process mono and RGB files, either can be chosen - Siril will autodetect the other one if
  both are installed. Note that for these old installations, the original executable names **rgb_starnet++**
  and **mono_starnet++** MUST be kept. For all newer single-executable versions of StarNet, Siril will
  determine the version heuristically and interface with it accordingly.

- **StarNet Weights Location**: New Torch-based versions of StarNet provide the option to provide the
  location of a neural network weights file: it need not be in the same directory as the executable. This
  preference can be used to set the location of a weights file to pass to StarNet, and it can be reset using
  the associated button. Note: this option only works with Torch-based StarNet installations. With older
  StarNet installations the weights file must be in the same directory as the executable.

  **Предупреждение**: This is the location of the command line version of StarNet that need to be given,
  not the GUI one.

- **Gnuplot Installation Directory**: In order to use lightcurve feature of Siril, it is necessary to install
  gnuplot. Then, you need to tell to Siril the path where gnuplot is located. On Unix-like systems, if the
  installation directory is in the **PATH** environment variable, it is not necessary.

  **Предупреждение**: On macOS, it can be difficult to find the directory path because Apple does not
  make browsing easy for some folders. A trick is to type **Shift + Cmd + g** on the open File Chooser
Рис. 14: Page 10 of preferences dialog
Dialog, then directly enter the installation path, which is usually the one set by Homebrew. Usually it is /usr/local/bin on intel computers and /opt/homebrew/bin/ on Apple Silicon versions.

- **Copyright TIFF**: When saving TIFF files it is possible to customize the copyright of the dedicated EXIF metadata.
- **Update**: By default Siril checks updates at startup. You are free to disable this behavior if you don’t want the application queries our website.

### 3.2 Настройки (команды)

Начиная с версии 1.2, большинство настроек могут быть установлены командами, что подразумевает прямой ввод команд в командной строке, либо в сценариях или в автономном режиме.

Для получения списка всех доступных переменных, введите в командной строке siril:

```
get -A
```

Это выведет список всех переменных с их текущим значением и кратким описанием в журнал (используйте опцию в нижнем регистре `-a`, чтобы опустить описание).

Этот список представлен в таблице ниже:

<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>core.wd</td>
<td>(не установлено)</td>
<td>директория</td>
<td>текущая рабочая директория</td>
</tr>
<tr>
<td>core.extension</td>
<td>.fit</td>
<td>строка</td>
<td>расширение файла FITS</td>
</tr>
<tr>
<td>core.force_16bit</td>
<td>false</td>
<td>логический</td>
<td>не используются 32 бита для разрядности</td>
</tr>
<tr>
<td>core.allow_heterogeneous_fits</td>
<td>false</td>
<td>логический</td>
<td>позволяет кубам FITS иметь разные размеры</td>
</tr>
<tr>
<td>core.mem_mode</td>
<td>0 [0, 1]</td>
<td>целочисленный</td>
<td>режим памяти (0 — отношение, 1 — объём)</td>
</tr>
<tr>
<td>core.mem_ratio</td>
<td>0.9 [0.05, 4]</td>
<td>двойной точности</td>
<td>доля доступной памяти</td>
</tr>
<tr>
<td>core.mem_amount</td>
<td>10 [0.1, 1e+06]</td>
<td>двойной точности</td>
<td>объём памяти в Гб</td>
</tr>
<tr>
<td>core.hd_bitdepth</td>
<td>20 [17, 24]</td>
<td>целочисленный</td>
<td>Разрядность расширенного авторастягивания</td>
</tr>
<tr>
<td>core.script_check_requires</td>
<td>true</td>
<td>логический</td>
<td>в сценарии необходима команда requires</td>
</tr>
<tr>
<td>core.pipe_check_requires</td>
<td>false</td>
<td>логический</td>
<td>в канале необходима команда requires</td>
</tr>
</tbody>
</table>

continues on next page
### Таблица 1 — продолжение с предыдущей страницы

<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>core.check_updates</td>
<td>true</td>
<td>логический</td>
<td>проверка обновлений при запуске</td>
</tr>
<tr>
<td>core.lang</td>
<td>(не установлено)</td>
<td>строка</td>
<td>активный язык siril</td>
</tr>
<tr>
<td>core.swap_dir</td>
<td>зависит от ОС</td>
<td>директория</td>
<td>директория подкачки</td>
</tr>
<tr>
<td>core.wcs_formalism</td>
<td>1 [0, 1]</td>
<td>целочисленный</td>
<td>формализм WCS, используемый в заголовке FITS</td>
</tr>
<tr>
<td>core.catalogue_namedstars</td>
<td>(*)</td>
<td>строка</td>
<td>Путь к каталогу namedstars.dat</td>
</tr>
<tr>
<td>core.catalogue_unnamedstars</td>
<td>(*)</td>
<td>строка</td>
<td>Путь к каталогу unnamedstars.dat</td>
</tr>
<tr>
<td>core.catalogue_tycho2</td>
<td>(*)</td>
<td>строка</td>
<td>Путь к каталогу deepstars.dat</td>
</tr>
<tr>
<td>core.catalogue_nomad</td>
<td>(*)</td>
<td>строка</td>
<td>Путь к каталогу USNO-NOMAD-1e8.dat</td>
</tr>
<tr>
<td>core.rgb_aladin</td>
<td>false</td>
<td>логический</td>
<td>добавить CTYPE3='RGB' в заголовок FITS</td>
</tr>
<tr>
<td>core.copyright</td>
<td>(не установлено)</td>
<td>строка</td>
<td>авторские права пользователя для размещения в заголовке файла</td>
</tr>
<tr>
<td>core.starnet_exe</td>
<td>(не установлено)</td>
<td>строка</td>
<td>расположение исполняемого файла StarNet</td>
</tr>
<tr>
<td>core.starnet_weights</td>
<td>(не установлено)</td>
<td>строка</td>
<td>расположение файла с весами StarNet-torch</td>
</tr>
<tr>
<td>core.gnuplot_dir</td>
<td>(не установлено)</td>
<td>строка</td>
<td>директория установки gnuplot</td>
</tr>
<tr>
<td>core.asnet_dir</td>
<td>(не установлено)</td>
<td>строка</td>
<td>директория установки asnet_ansvr</td>
</tr>
<tr>
<td>core.fftw_timelimit</td>
<td>60</td>
<td>двойной точности</td>
<td>Ограничение времени планирования FFTW</td>
</tr>
<tr>
<td>core.fftw_conv_fft_cutoff</td>
<td>15</td>
<td>целочисленный</td>
<td>Минимальный размер ядра свёртки для использования FFTW</td>
</tr>
<tr>
<td>core.fftw_strategy</td>
<td>1</td>
<td>целочисленный</td>
<td>Стратегия планирования FFTW</td>
</tr>
<tr>
<td>мультипотоковое FFTW</td>
<td>true</td>
<td>логический</td>
<td>мультипотоковое FFTW</td>
</tr>
<tr>
<td>starfinder.focal_length</td>
<td>0 [0, 999999]</td>
<td>двойной точности</td>
<td>фокусное расстояние в мм для настройки радиуса</td>
</tr>
<tr>
<td>starfinder.pixel_size</td>
<td>0 [0, 99]</td>
<td>двойной точности</td>
<td>размер пикселя в мкм для настройки радиуса</td>
</tr>
<tr>
<td>debayer.use_bayer_header</td>
<td>true</td>
<td>логический</td>
<td>шаблон из заголовка файла</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>debayer.pattern</td>
<td>0 [0, 7]</td>
<td>целочисленный</td>
<td>индекс шаблона Байера</td>
</tr>
<tr>
<td>debayer.interpolation</td>
<td>8 [0, 10]</td>
<td>целочисленный</td>
<td>тип интерполяции</td>
</tr>
<tr>
<td>debayer.top_down</td>
<td>true</td>
<td>логический</td>
<td>принудительная дебайеризация сверху вниз</td>
</tr>
<tr>
<td>debayer.offset_x</td>
<td>0 [0, 1]</td>
<td>целочисленный</td>
<td>Смещение шаблона Байера по X</td>
</tr>
<tr>
<td>debayer.offset_y</td>
<td>0 [0, 1]</td>
<td>целочисленный</td>
<td>Смещение шаблона Байера по Y</td>
</tr>
<tr>
<td>debayer.xtrans_passes</td>
<td>1 [1, 4]</td>
<td>целочисленный</td>
<td>Число проходов для алгоритма X-Trans Markesteijn</td>
</tr>
<tr>
<td>photometry.gain</td>
<td>2.3 [0, 10]</td>
<td>двойной точности</td>
<td>электронов/ADU для оценки шума</td>
</tr>
<tr>
<td>photometry.inner</td>
<td>20 [2, 100]</td>
<td>двойной точности</td>
<td>внутренний радиус кольца фона</td>
</tr>
<tr>
<td>photometry.outer</td>
<td>30 [3, 200]</td>
<td>двойной точности</td>
<td>внешний радиус кольца фона</td>
</tr>
<tr>
<td>photometry.inner_factor</td>
<td>4.2 [2, 50]</td>
<td>двойной точности</td>
<td>коэффициент для автоматического расчёта внутреннего радиуса</td>
</tr>
<tr>
<td>photometry.outer_factor</td>
<td>6.3 [2, 50]</td>
<td>двойной точности</td>
<td>коэффициент для автоматического расчёта внешнего радиуса</td>
</tr>
<tr>
<td>photometry.force_radius</td>
<td>true</td>
<td>логический</td>
<td>значение апертуры светового потока</td>
</tr>
<tr>
<td>photometry.aperture</td>
<td>10 [1, 100]</td>
<td>двойной точности</td>
<td>значение апертуры для вычисления светового потока</td>
</tr>
<tr>
<td>photometry.minval</td>
<td>-1500 [-65536, 65534]</td>
<td>двойной точности</td>
<td>минимально надёжное значение пикселя для фотометрии</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>photometry.maxval</td>
<td>60000 [1, 65535]</td>
<td>двойной точности</td>
<td>максимально надёжное значение пикселя для фотометрии</td>
</tr>
<tr>
<td>astrometry.asnet_percent_scale_range</td>
<td>20 [0, 10000]</td>
<td>целочисленный</td>
<td>процент, определяющий допустимые нижнюю и верхнюю границы масштаба</td>
</tr>
<tr>
<td>astrometry.asnet_sip_order</td>
<td>0 [0, 6]</td>
<td>целочисленный</td>
<td>степень полиномиальной коррекции</td>
</tr>
<tr>
<td>astrometry.asnet_radius</td>
<td>10 [0.01, 180]</td>
<td>двойной точности</td>
<td>радиус вокруг координат цели (градусы)</td>
</tr>
<tr>
<td>astrometry.asnet_keep_xyls</td>
<td>false</td>
<td>логичееский</td>
<td>не удалять таблицы .xyls</td>
</tr>
<tr>
<td>astrometry.asnet_keepwcs</td>
<td>false</td>
<td>логичееский</td>
<td>не удалять файлы .wcs</td>
</tr>
<tr>
<td>astrometry.asnet_max_seconds</td>
<td>10 [0, 100000]</td>
<td>целочисленный</td>
<td>максимальное кол-во секунд для попытки решения</td>
</tr>
<tr>
<td>astrometry.asnet_show_output</td>
<td>false</td>
<td>логичееский</td>
<td>показать вывод solve-field в журнале</td>
</tr>
<tr>
<td>astrometry.update_default_scale</td>
<td>true</td>
<td>логичееский</td>
<td>обновить фокусное расстояние и размер пикселя, используемые по умолчанию, из результатов</td>
</tr>
<tr>
<td>analysis.panel</td>
<td>256 [127, 1024]</td>
<td>целочисленный</td>
<td>размер панели диалога аберраций</td>
</tr>
<tr>
<td>analysis.window</td>
<td>381 [300, 1600]</td>
<td>целочисленный</td>
<td>размер окна диалога аберраций</td>
</tr>
<tr>
<td>compression.enabled</td>
<td>false</td>
<td>логичееский</td>
<td>Включено сжатие FITS</td>
</tr>
<tr>
<td>compression.method</td>
<td>0 [0, 3]</td>
<td>целочисленный</td>
<td>Метод сжатия FITS</td>
</tr>
<tr>
<td>compression.quantization</td>
<td>16 [8, 256]</td>
<td>двойной точности</td>
<td>коэффициент квантования для 32 битных значений с плавающей запятой</td>
</tr>
<tr>
<td>compression.hcompress_scale</td>
<td>4 [0, 256]</td>
<td>двойной точности</td>
<td>Коэффициент масштаба HCompress</td>
</tr>
<tr>
<td>gui_prepro.cfa</td>
<td>false</td>
<td>логичееский</td>
<td>тип сенсора для косметической коррекции</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию (Диапазон)</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>gui_prepro.equalize_cfa</td>
<td>true</td>
<td>логический</td>
<td>выравнять плоские каналы</td>
</tr>
<tr>
<td>gui_prepro.fix_xtrans</td>
<td>false</td>
<td>логический</td>
<td>включить коррекцию для сенсора X-Trans</td>
</tr>
<tr>
<td>gui_prepro.xtrans_af_x</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_af_y</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_af_w</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_af_h</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_sample_x</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_sample_y</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_sample_w</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.xtrans_sample_h</td>
<td>0</td>
<td>целочисленный</td>
<td>если никакой модели X-Trans не найдено, использовать</td>
</tr>
<tr>
<td>gui_prepro.bias_lib</td>
<td>(не установлено)</td>
<td>строка</td>
<td>мастер-кадр шума считывания по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.use_bias_lib</td>
<td>false</td>
<td>логический</td>
<td>использовать мастер-кадр шума считывания по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.dark_lib</td>
<td>(не установлено)</td>
<td>строка</td>
<td>темновой мастер-кадр по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.use_dark_lib</td>
<td>false</td>
<td>логический</td>
<td>использовать темновой мастер-кадр по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.flat_lib</td>
<td>(не установлено)</td>
<td>строка</td>
<td>мастер-кадр плоского поля по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.use_flat_lib</td>
<td>false</td>
<td>логический</td>
<td>использовать мастер-кадр плоского поля по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.stack_default</td>
<td>$seqname$stacked</td>
<td>строка</td>
<td>название стопки по умолчанию</td>
</tr>
<tr>
<td>gui_prepro.use_stack_default</td>
<td>true</td>
<td>логический</td>
<td>использовать предпочтительное название стека</td>
</tr>
<tr>
<td>gui_registration.method</td>
<td>0 [0, 7]</td>
<td>целочисленный</td>
<td>индекс выбранного метода регистрации</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>gui_registration.interpolation</td>
<td>4 [0, 5]</td>
<td>целочисленный</td>
<td>индекс выбранного метода интерполяции</td>
</tr>
<tr>
<td>gui_registration.clamping</td>
<td>true</td>
<td>логический</td>
<td>использование метода фиксации для интерполяции методом Ланцош-4 или бикубической интерполяции</td>
</tr>
<tr>
<td>gui_stack.method</td>
<td>0 [0, 4]</td>
<td>целочисленный</td>
<td>индекс выбранного метода</td>
</tr>
<tr>
<td>gui_stack.normalization</td>
<td>3 [0, 4]</td>
<td>целочисленный</td>
<td>индекс метода нормализации</td>
</tr>
<tr>
<td>gui_stack.rejection</td>
<td>5 [0, 7]</td>
<td>целочисленный</td>
<td>индекс метода выбраковки</td>
</tr>
<tr>
<td>gui_stack.weighting</td>
<td>0 [0, 4]</td>
<td>целочисленный</td>
<td>индекс метода присвоения веса</td>
</tr>
<tr>
<td>gui_stack.sigma_low</td>
<td>3 [0, 20]</td>
<td>двойной точности</td>
<td>нижнее значение для выбраковки</td>
</tr>
<tr>
<td>gui_stack.sigma_high</td>
<td>3 [0, 20]</td>
<td>двойной точности</td>
<td>верхнее значение для выбраковки</td>
</tr>
<tr>
<td>gui_stack.linear_low</td>
<td>5 [0, 20]</td>
<td>двойной точности</td>
<td>значение нижнего порога для выбраковки</td>
</tr>
<tr>
<td>gui_stack.linear_high</td>
<td>5 [0, 20]</td>
<td>двойной точности</td>
<td>значение верхнего порога для выбраковки</td>
</tr>
<tr>
<td>gui_stack.percentile_low</td>
<td>3 [0, 100]</td>
<td>двойной точности</td>
<td>нижнее значение процентиля для выбраковки</td>
</tr>
<tr>
<td>gui_stack.percentile_high</td>
<td>3 [0, 100]</td>
<td>двойной точности</td>
<td>верхнее значение процентиля для выбраковки</td>
</tr>
<tr>
<td>gui.first_start</td>
<td>(не установлено)</td>
<td>строка</td>
<td>первый запуск siril</td>
</tr>
<tr>
<td>gui.silent_quit</td>
<td>false</td>
<td>логический</td>
<td>не спрашивать перед выходом</td>
</tr>
<tr>
<td>gui.silent_linear</td>
<td>false</td>
<td>логический</td>
<td>не спрашивать перед сохранением в нелинейном режиме</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>gui.remember_windows</td>
<td>true</td>
<td>логический</td>
<td>запоминать положение окна</td>
</tr>
<tr>
<td>gui.main_win_pos_x</td>
<td>0</td>
<td>целочисленный</td>
<td>положение главного окна</td>
</tr>
<tr>
<td>gui.main_win_pos_y</td>
<td>0</td>
<td>целочисленный</td>
<td>положение главного окна</td>
</tr>
<tr>
<td>gui.main_win_pos_w</td>
<td>0</td>
<td>целочисленный</td>
<td>положение главного окна</td>
</tr>
<tr>
<td>gui.main_win_pos_h</td>
<td>0</td>
<td>целочисленный</td>
<td>положение главного окна</td>
</tr>
<tr>
<td>gui.pan_position</td>
<td>-1</td>
<td>целочисленный</td>
<td>положение двухстороннего разделителя</td>
</tr>
<tr>
<td>gui.extended</td>
<td>true</td>
<td>логический</td>
<td>главное окно расширено</td>
</tr>
<tr>
<td>gui.maximized</td>
<td>false</td>
<td>логический</td>
<td>главное окно развёрнуто</td>
</tr>
<tr>
<td>gui.theme</td>
<td>0 [0, 1]</td>
<td>целочисленный</td>
<td>индекс выбранной темы</td>
</tr>
<tr>
<td>gui.font_scale</td>
<td>100</td>
<td>двойной точности</td>
<td>масштаб шрифта в процентах</td>
</tr>
<tr>
<td>gui.icon_symbolic</td>
<td>false</td>
<td>логический</td>
<td>стиль иконок</td>
</tr>
<tr>
<td>gui.script_path</td>
<td></td>
<td>список строк</td>
<td>список каталогов со сценариями</td>
</tr>
<tr>
<td>gui.warn_script_run</td>
<td>true</td>
<td>логический</td>
<td>предупреждать при запуске сценария</td>
</tr>
<tr>
<td>gui.show_thumbnails</td>
<td>true</td>
<td>логический</td>
<td>показывать миниатюры при выборе файлов</td>
</tr>
<tr>
<td>gui.thumbnail_size</td>
<td>256</td>
<td>целочисленный</td>
<td>размер миниатюр</td>
</tr>
<tr>
<td>gui.selection_guides</td>
<td>0</td>
<td>целочисленный</td>
<td>количество элементов направляющих сетки</td>
</tr>
<tr>
<td>gui.show_deciasec</td>
<td>false</td>
<td>логический</td>
<td>показать десятые доли угловых секунд при наведении</td>
</tr>
<tr>
<td>gui.default_rendering_mode</td>
<td>0 [0, 6]</td>
<td>целочисленный</td>
<td>режим показа по умолчанию</td>
</tr>
</tbody>
</table>

continues on next page
<table>
<thead>
<tr>
<th>Переменная</th>
<th>По умолчанию ([Диапазон])</th>
<th>Тип</th>
<th>Комментарий</th>
</tr>
</thead>
<tbody>
<tr>
<td>gui.display_histogram_mode</td>
<td>0 [0, 1]</td>
<td>целочисленный</td>
<td>режим показа гистограммы по умолчанию</td>
</tr>
<tr>
<td>gui.astrometry.compass_position</td>
<td>1 [0, 5]</td>
<td>целочисленный</td>
<td>индекс положения компаса над сеткой</td>
</tr>
<tr>
<td>gui.astrometry.cat_messier</td>
<td>true</td>
<td>логический</td>
<td>показывать объекты каталога Мессье в аннотациях</td>
</tr>
<tr>
<td>gui.astrometry.cat_ngc</td>
<td>true</td>
<td>логический</td>
<td>показывать объекты каталога NGC в аннотациях</td>
</tr>
<tr>
<td>gui.astrometry.cat_ic</td>
<td>true</td>
<td>логический</td>
<td>показывать объекты каталога IC в аннотациях</td>
</tr>
<tr>
<td>gui.astrometry.cat_ldn</td>
<td>true</td>
<td>логический</td>
<td>показывать объекты каталога LDN в аннотациях</td>
</tr>
<tr>
<td>gui.astrometry.cat_sh2</td>
<td>true</td>
<td>логический</td>
<td>показывать объекты каталога SH2 в аннотациях</td>
</tr>
<tr>
<td>gui.astrometry.cat_stars</td>
<td>true</td>
<td>логический</td>
<td>показать звёзды в аннотациях</td>
</tr>
<tr>
<td>gui.astrometry.cat_user</td>
<td>true</td>
<td>логический</td>
<td>показать пользовательские объекты в аннотациях</td>
</tr>
<tr>
<td>gui.pixelmath.pm_presets</td>
<td>список строк</td>
<td>список предустановок математических операций</td>
<td></td>
</tr>
</tbody>
</table>

(*) Для каталогов kstars значением по умолчанию будет `~/.local/share/kstars/`, независимо от используемой ОС.

В любом случае вам нужно будет скачать их и указать место их расположения.

См. раздел использование локальных каталогов.

Значения можно получить с помощью команды get:

Командная строка Siril

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

Получает значение переменной, используя её имя или отображает все переменные с помощью -a (имя и значение) или -A (подробный список)

Значения можно изменить с помощью команды set:

Командная строка Siril
Обновляет значение настройки, используя имя её переменной с заданным значением или набором значений, используя существующий ini-файл с опцией -import=.
См. get для получения списка переменных или их значений
Глава 4

Форматы файлов

Siril может открывать и работать с рядом форматов файлов. Однако чтение только двух форматов поддерживается нативно и позволяет создавать последовательности: FITS и SER.

Ниже приведено описание различных форматов файлов, чтение которых поддерживается в Siril, дана информация об ограничении некоторых форматов файлов и сильных сторонах других.

4.1 Разрядность

Разрядность определяет число битов, используемых для указания цвета отдельного пикселя или число битов, используемых для каждой цветовой компоненты отдельного пикселя.

Для повседневного использования 8 битных изображений, в которых для кодирования пикселя в изображении используются значения в диапазоне [0, 255], более чем достаточно. Однако фотографирование астрономических объектов более требовательно и обычно при этом работа идёт с изображениями с разрядностью по крайней мере 16 бит, m.e. в диапазоне [0, 65535]. Более того, 32 битная разрядность позволяет сохранить едва различимые данные. В последнем случае пиксели кодируются в диапазоне [0, 4294967295], или, как это имеет место в Siril, дробными значениями в диапазоне [0, 1]. Возможно найти форматы, кодирующие пиксели в 64 битах (в диапазоне [0, 1]), но такие форматы редки и имеют специфические области применения. В частности, формат FITS позволяет это.

Однако не все форматы изображений поддерживают 16 битную разрядность, не говоря уже о 32 битной. Это следует иметь в виду, выбирая формат для работы.
Рис. 1: Линейное изображение, сохранённое с разрядностью 16 бит
Рис. 2: Это же изображение, сохранённое с разрядностью 8 бит. Практически все данные потеряны
4.2 Common File Formats

The image file formats presented here are standard formats, readable by all image manipulation software. These formats were designed to meet the needs some time ago and may be obsolete. Moreover, none of these formats have been designed to handle astronomical data. They must therefore generally be used at the end of the processing chain.

4.2.1 BMP

Files with the .bmp extension are bitmap image files used to store digital bitmap images. These images are independent of the graphics card and are also called Device Independent Bitmap (DIB) file format. This independence allows the file to be opened on multiple platforms such as Microsoft Windows and Mac. The BMP file format allows data to be stored as two-dimensional digital images, both in monochrome and in color, with different color depths.

Nowadays, this format is not really used anymore and other file types are preferred.

4.2.2 JPEG

Probably the most used file format for sharing images on forums, by e-mail or usb sticks. This format allows a more or less strong (destructive) compression which gives ideal file sizes for exchanges. The extension of this type of file is .jpg or .jpeg.

The JPEG format is however only coded in 8-bit. With the compression that produces artifacts, this format is not very suitable for astronomy images and we generally prefer the PNG format.

4.2.3 PNG

Portable Network Graphics is a raster-graphics file format that supports lossless data compression. The extension of the format is .png. PNG grayscale images support the widest range of pixel depths of any image type. Depths of 1, 2, 4, 8, and 16-bit are supported, covering everything from simple black-and-white scans to full-depth medical and raw astronomical images.

Calibrated astronomical image data is usually stored as 32-bit or 64-bit floating-point values, and some raw data is represented as 32-bit integers. Neither format is directly supported by PNG.

However, this format is an excellent choice for saving the final image, after processing.

4.2.4 TIFF

TIFF or TIF, Tagged Image File Format, represents raster images for use on various devices that conform to this file format standard. It is capable of describing bi-level, grayscale, palette color and full color image data in multiple color spaces. It supports both lossless and lossy compression schemes to allow applications that use this format to choose between space and time. The extension is either .tiff or .tif.

The advantages of the TIFF format are multiple. It supports encoding up to 32-bit per pixel and offers a wide variety of possible fields in the metadata making it a good candidate for storing astronomical data.

Using the TIFF format, and in collaboration with other developers, we have set up a pseudo standard, Astro-TIFF.
4.2.5 NetPBM

Several graphics formats are used and defined by the Netpbm project. The portable pixmap format (PPM), the portable graymap format (PGM) and the portable bitmap format (PBM) are image file formats designed to be easily exchanged between platforms. Possible file extension are .pbm, .pgm (for gray scale files) and .ppm.

These formats, supporting up to 16-bit per channel, are marginally used and should only be used for final image storage.

4.2.6 AVI

It's a film container, able to contain data with various audio and video codecs. Some lossless video codecs exist and they have been used for astronomy imaging purposes in the past, but it's a format that does not contain metadata usable for astronomy, that is limited to 8-bit images and that does not give any warranty that the data it contains is raw.

Предупреждение: This input file format is now deprecated. We recommend to use SER format instead.

4.3 FITS

4.3.1 Спецификация

FITS stands for Flexible Image Transport System and is the standard astronomical data format used by professional scientists such as NASA. FITS is much more than an image format (such as JPG or TIFF) and is primarily designed to store scientific data consisting of multidimensional arrays.

A FITS file consists of one or more header and data units (HDUs), with the first HDU referred to as the "primary HDU" or "primary array." Five primary data types are supported: 8-bit unsigned bytes, 16 and 32-bit signed integers, and 32 and 64-bit single and double-precision floating-point reals. The FITS format can also store 16 and 32-bit unsigned integers.

Each header unit consists of any number of 80-character keyword records which have the general form:

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

The keyword names may be up to 8 characters long and can only contain uppercase letters, the digits 0-9, the hyphen, and the underscore character. The keyword name is (usually) followed by an equals sign and a space character (= ) in columns 9 - 10 of the record, followed by the value of the keyword which may be either an integer, a floating point number, a character string (enclosed in single quotes), or a boolean value (the letter T or F).

The last keyword in the header is always the END keyword which has no value or comment fields.

Each header unit begins with a series of required keywords that specify the size and format of the following data unit. A 2-dimensional image primary array header, for example, begins with the following keywords:

```
SIMPLE = T / file does conform to FITS standard
BITPIX = 16 / number of bits per data pixel
NAXIS = 2 / number of data axes
```

(continues on next page)
4.3.2 Сжатие

Compression is the way to reduce the size of the image. There are many methods of compression depending on the type of images used. This compression can be destructive, as with the JPEG, or lossless as proposed by the PNG.

It is possible to work with compressed FITS files. At the cost of a longer calculation time, the size of the images can be reduced considerably. Siril offers several compression algorithms which are the following:

- **Rice**: The Rice algorithm is simple and very fast
- **GZIP 1**: The gzip algorithm is used to compress and uncompress the image pixels. Gzip is the compression algorithm used in the free GNU software utility of the same name.
- **GZIP 2**: The bytes in the array of image pixel values are shuffled into decreasing order of significance before being compressed with the gzip algorithm. This is usually especially effective when compressing floating-point arrays.

С этими алгоритмами связана одна опция, **Уровень квантования**:

В то время как изображения в формате с плавающей точкой могут быть сжаты без потерь (используя gzip, поскольку алгоритм Райса сжимает только целочисленные массивы), эти изображения часто сжимаются не очень хорошо, потому что значения пикселей слишком зашумлены; менее значимые биты в мантиссе значений пикселей эффективно содержат несжимаемые случайные битовые шаблоны. Чтобы добиться более высокого сжатия, необходимо удалить часть этого шума, но без потери полезной информации. Если он слишком велик, один из них неправильно выбирает значения пикселей, что приводит к потере информации в изображении. Однако, если он слишком мал, он сохраняет слишком много шума (или даже усиливает его) в значениях пикселей, что приводит к плохому сжатию.

Примечание:  Поддерживаемые алгоритмы обеспечивают сжатие изображений без потерь (loss-less), когда применяются к целочисленным изображениям FITS; значения пикселей сохраняются без потери информации в ходе сжатия и распаковки. Изображения FITS с плавающей запятой (имеющие **BITPIX** = -32 или -64) сначала квантуются до масштабированных целочисленных значений, а потом сжимаются. Этот способ обеспечивает гораздо более высокие коэффициенты сжатия, что чем сжатие изображения с использованием GZIP. Однако это означает, что исходные значения с плавающей запятой не могут точно восстановлены при распаковке изображения. При правильном применении, это только удаляет 'шум' из значений с плавающей запятой без потери значимой информации.
4.3.3 Ориентация изображений FITS

The FITS standard is a container that describes how to store image data and metadata. Professional tools, from the early age of the FITS format, like ds9 (Harvard Smithsonian Center for Astrophysics), fv (FITS viewer from NASA), store images bottom-up. We might be tempted to say that it does not really matter, but when demosaicing or astrometry is involved, problems arise. For example, the usual RGGB Bayer pattern becomes GBRG if the image is upside-down.

Nowadays, despite this, most camera drivers are writing data in the top-down order and we have to cope with it.

For these reasons, we recently have introduced, together with P. Chevalley of CCDCiel, a new FITS keyword. We encourage all data producers, INDI and ASCOM developers, to use it in order to make things easier for everybody.

This keyword is ROWORDER of type TSTRING. It can take two values: BOTTOM-UP and TOP-DOWN.

Siril will always read and display images in the bottom-up order, however if the top-down information is specified in the keyword, then Siril will demosaic the image with the corrected pattern.

Why would some programs write images bottom-up in the first place?

The reason is: mathematics do it that way. Also, the FITS specification says:

5.1. Image display conventions

It is very helpful to adopt a convention for the display of images transferred via the FITS format. Many of the current image processing systems have converged upon such a convention. Therefore, we recommend that FITS writers order the pixels so that the first pixel in the FITS file (for each image plane) be the one that would be displayed in the lower-left corner (with the first axis increasing to the right and the second axis increasing upwards) by the imaging system of the FITS writer. This convention is clearly helpful in the absence of a description of the world coordinates. It does not preclude a program from looking at the axis descriptions and overriding this convention, or preclude the user from requesting a different display. This convention also does not excuse FITS writers from providing complete and correct descriptions of the image coordinates, allowing the user to determine the meaning of the image. The ordering of the image for display is simply a convention of convenience, whereas the coordinates of the pixels are part of the physics of the observation.

Предупреждение: ROWORDER keyword can be used for:

1. Displaying the image with the intended orientation (unflip the display).
2. Unflip the Bayer demosaic pattern. So the demosaic pattern can be specified conform the sensor supplier.

BUT

1. ROWORDER shall not be used to unflip the image data for stacking. Otherwise new images would become incompatible with older darks and flats.
2. ROWORDER shall not be used to unflip the image data for astrometric solving. This would make the astrometric solution incompatible with other programs.
4.3.4 Программное обеспечение, использующее это ключевое слово

- Siril (начиная с версии 0.99.4)
- CCDCiel (начиная с версии 0.9.72)
- Indi (с июля 2020)
- KStars (начиная с 3.4.3)
- SharpCap (начиная с версии 3.3)
- FireCapture (начиная с версии 2.7)
- N.I.N.A (начиная с версии 1.10)
- MaxImDL (начиная с версии 6.23)
- INDIGO (с июля 2020)
- PixInsight (начиная с версии 1.8.8-6)
- ASTAP (начиная с версии 80.9.391)
- APT (начиная с версии 3.86.3)
- AstroDMx Capture (начиная с версии 0.80)
- Astroart (начиная с версии 8.0)

4.3.5 Retrieving the Bayer matrix

Image row order changes the way the Bayer matrix should be read, but there are also two optional FITS header keywords that have an effect on this: XBAYROFF and YBAYROFF. They specify an offset to the Bayer matrix, to start reading it on first column or first row.

To help developers integrating the ROWORDER, XBAYROFF and YBAYROFF keywords in their software, some test images were created by Han Kleijn from hnsky.org, one for each combination of the three keywords. Download them here: Bayer_test_pattern_v6.tar.gz.

4.4 Astro-TIFF

In 2022, Han Kleijn, developer of the ASTAP software, offered to contribute to the development of a new pseudo-standard using the TIFF format and taking advantage of the power of FITS file headers. This is how the Astro-TIFF format was born.

4.4.1 Why a new standard among all the others?

Currently, the most used format for astrophotography is the FITS format. This one, developed by processional scientists, meets all the expectations of amateurs. And although its great flexibility causes some concerns of compatibility it remains the format to be preferred.

Other specialized formats exist but are usually associated with a specific software. Like the XISF format developed by the PixInsight team. This last format, although often requested in Siril, is a format dedicated to PixInsight, a proprietary software. The interest of developing a compatibility with Siril is therefore minimal and we do not wish to introduce it. If one day a XISF library exists, maybe we will change our mind.

Developing Astro-TIFF appears then as a good alternative, because based on the TIFF format, it is possible to open the files on any image processing software.
Finally, the TIFF format supports compression (as does the FITS format) which allows for smaller image sizes.

### 4.4.2 Specification 1.0

**Dated 2022-06-21**

- Files are following the TIFF 6.0 specification including supplement 2 fully.
- The FITS header is written to the TIFF baseline tag **Image Description**. Code 270, Hex 010E.
- The header is following the FITS specification except that the lines can be shorter then 80 characters and lines are ending with either CR+LF (0D0A) or LF (0A).
- First line in the description is the first header line and starts with **SIMPLE**. The last line of the header starts with **END**.

**Recommendations**

- **TIFFtag_orientation=1** (left-top) Orientation is following the conventions. Pixel **FITS_image[1,1]** is left-bottom. **TIFF_image[0,0]** is left-top. These pixels are first written or read from the file. So when writing a FITS image into TIFF preserving the orientation for the user, the first pixel to write is **FITS_image[1,NAXIS2]**.
- **TIFFtag_compression=8** (Deflate) or 5 (LZW).
- For greyscale images **TIFFtag_PhotometricInterpretation = 1** (minimum value is black, maximum is white).
- Write all available header keywords.

**Notes**

- This use of TIFF format is intended for 16-bit lights, darks, flats and flat-darks (astronomical images), but can also be used in the 32-bit format. It is possible to convert FITS to TIFF and backwards but the application programmer can decide to export only (write) or only import (read) in Astro-TIFF format.
- If an astrometrical (plate) solution is included then it should match with the image orientation.
- Some header keywords are redundant like **NAXIS1, NAXIS2, BZERO** and **BITPIX** and are not required. TIFF image dimensions and type are leading.
- The de-mosaic pattern specified in the header should match with the image orientation.
- The header will be visible in many image manipulation programs.

**Example of an Astro-TIFF header that looks just like a FITS file header:**

<table>
<thead>
<tr>
<th>SIMPLE</th>
<th>T / file does conform to FITS standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>BITPIX</td>
<td>-32 / number of bits per data pixel</td>
</tr>
<tr>
<td>NAXIS</td>
<td>2 / number of data axes</td>
</tr>
<tr>
<td>NAXIS1</td>
<td>6248 / length of data axis 1</td>
</tr>
<tr>
<td>NAXIS2</td>
<td>4176 / length of data axis 2</td>
</tr>
<tr>
<td>NAXIS3</td>
<td>1 / length of data axis 3</td>
</tr>
<tr>
<td>EXTEND</td>
<td>T / FITS dataset may contain extensions</td>
</tr>
</tbody>
</table>
COMMENT FITS (Flexible Image Transport System) format is defined in 'Astronomy 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

OBERVER = '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

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)

PLTSOLV = T / Siril internal solve

HISTORY Background extraction (Correction: Subtraction)

HISTORY Plate Solve

END
4.4.3 Saving Astro-TIFF in Siril

In Siril you can save Astro-TIFF files by choosing the TIFF format in the save dialog when you click on Save As. The drop-down list in the TIFF dialog allows you to choose between saving in standard TIFF format or in Astro-TIFF format. The latter is the default format.

![Save Dialog with Astro-TIFF option](image)

Командная строка Siril

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

Saves current image under the form of a uncompressed TIFF file with 16-bit per channel: `filename.tif`. The option `-astro` allows saving in Astro-tiff format, while `-deflate` enables compression.
4.4.4 Sample files

- Astro-TIFF file created by Siril (32-bit, uncompressed).
- Astro-TIFF file created by Siril (32-bit, compressed).

4.5 SER

SER file format is a simple image sequence format, similar to uncompressed films. Documentation can be found on the official page. The latest PDF document is mirrored on free-astro too.

With improvements of version 2 and 3, SER handles colour images, which makes it perfect as replacement for the usual AVI or other film format produced by older capture programs in all astronomy situations. Compressed images should not be used for astronomy but can still be converted to SER, which will make the files bigger for the same quality, but easier to work with.

Siril can convert any image sequence and many film formats into SER files. Ser-player is a great tool that allows SER files to be visualised just like any film, with many options and works on most operating systems.

The main issue with AVI and other film containers is that it is designed to work with many codecs and pixel formats, which it good for general purpose films, but requires astronomy software to handle a large array of actually different file formats. General purpose film software are often not well equipped to handle 16-bit per pixel values or some uncompressed data formats. With SER, only one file format handles it all, that's why Siril for example is now developing processing only for SER.

4.5.1 File structure

A SER file has three parts:

- a 178-byte header containing images and observation information
- image data, raw pixel data
- an optional trailer containing dates for all images of the sequence

4.5.2 Handling colours

In version 3 (2014), there are two ways of handling coloured images in SER. If data comes directly from a sensor, the preferred way is probably to use one-plane images and interpolating data from the colour filter array (similarly to CFA file formats used in astronomy software).

The other way, added in version 3, is to use three planes to represent RGB image data. SER v3 supports RGB/BGR 8/16-bit data. This can be useful if data is converted from a source with an unknown colour filter array or for general purpose conversion.
4.5.3 Specification issue with endianness

Since SER files can contain 16-bit precision images, endianness must be well specified. The specification allows endianness to be either big-endian or little-endian, to facilitate file creation on various systems, as long as the used endianness is documented in the file's header.

For an unknown reason, several of the first programs to support SER disrespect the specification regarding the endianness flag. The specification states that a boolean value is used for the LittleEndian header, and they use it as a BigEndian header, with 0 for little-endian and 1 for big-endian. Consequently, to not break compatibility with these first implementations, later programs, like Siril, GoQat, Ser-player and many others, have also decided to implement this header in opposite meaning to the specification.
Sequences are what Siril uses to represent a set of manipulated files, for example the set of dark images that we'll turn into the master dark. It is a very useful tool for handling a large number of files that need to be linked with each other.

### 5.1 A set of two or more FITS files

Siril uses natively 32-bit floating point data or 16-bit unsigned integer data for the *FITS* images, other formats are automatically converted. To be recognized and detected as a sequence, FITS images file names must respect a particular pattern which is:

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

- `basename` can be anything using ascii characters. It is usually convenient, but not mandatory, to have it end with the `_` character. It will be used as the sequence name.
- `$i$` is the index of the image. It must be a positive number and can have several leading zeros.
- `[ext]` is the supported extension as explained in the *settings*, *fit* by default.

**Примечание:** The extension used to detect FITS sequences in the current working directory will be the same as the extension configured in the settings and as the files created by Siril.

**Предупреждение:** Some operating systems limit the number of images that can be opened at the same time, which is required for median or mean stacking methods. For Windows, the limit is 2048 images. If you have a lot of images, you should use another type of sequence, described below.
5.2 A single SER file

SER is a format meant to contain an acquisition sequence of several contiguous images in a single file. It is a rather simple format that cannot contain as much metadata as FITS, but more than simple films and data is not compressed. SER files can contain images of 8 or 16-bits per channel only. There are three types of SER files, depending on the pixel content: monochrome, CFA or color (3 channels).

Примечание: An SER file can be opened either via File and Open, or with the Search sequences button.

See here for more information on the SER format and why film formats like uncompressed AVI should not be used for astronomy.

Предупреждение: To some extent, a regular film file such as AVI or any other container are supported too. Film files support is being dropped in favour of SER, but it can still be useful to open a film in Siril, to explore its content, extract some frames or convert them. A few operations can still be done, but in a slower way than with other sequences, like sum stacking. For a complete processing you will face limitations and incompatibilities.

5.3 A single FITS file

Примечание: Also called FITS cubes or FITS sequences, or FITSEQ for short in Siril.

The FITS format is an image and science data container, it can contain several of these in a single file. We can use that to store an entire sequence of FITS images in a single file while preserving the FITS header of each image. It is the file format that professional astronomers use.

It’s simpler to manage one file on the disk than 2000, but since it is a single file, some operations on single images of the sequence may not be possible. In particular, it is not currently supported within Siril to change the header of a single image.

This format is an alternative to SER for a single-file sequence, with 32 bits per channel and full header support.

5.4 Загрузка последовательности

Рис. 1: Sequence Search and Cleaning.

When the working directory is set in the right place, the FITS follow the correct nomenclature, and the extension of the FITS files is also set correctly, then click the Search Sequence button of the Sequence
5.5 Frame selector

A great strength of Siril is that it easily manipulates image sequences. When a sequence is opened, the reference image (see below) will be displayed, by default this is the first image. However, it can sometimes be useful to inspect individual images of a sequence. This is possible with the frame selector, available via the toolbar with the button or via the sequence tab with the Open Frame List button.

![Frame selector](image)

Рис. 2: Frame selector that allows you to choose a frame from the sequence and display it, set it as reference or exclude it.

Clicking on an image in the list will load it and display it in the main area, while keeping the sequence as the active object for processing. More than just a image display selector, the tool can also be used to manually exclude images from the sequence, or visualize which are still included, visualize the values of image quality and shift between images if they have been computed, and change the reference image. Note that more image quality information can be viewed in the Plot tab.

Excluding an image from the sequence does not mean its data will be permanently deleted, it will just not be used for the subsequent processing operations, if instructed to do so. In most cases, the option to look for is called Process included images only.

The reference image is the image in the sequence that will serve as target for the registration and for the normalization. Other images will be transformed to look like the reference image, so it should be chosen carefully. Fortunately, since Siril 1.2, a new two-pass registration can automatically select the best image of the sequence as reference image before proceeding to image transformation.

The header bar of the window will provide many controls for these sequence properties:
- The drop-down menu allows the channel for which registration data (quality, shifts) is displayed to be changed, if they exist for other channels.
- The first button of the toolbar sets all images of the sequence as manually excluded.
- The second one, sets them all as included.

Командная строка Siril

```bash
select sequencename from to
```

This command allows easy mass selection of images in the sequence `sequencename` (from `from` to `to` included)

Командная строка Siril

```bash
unselect sequencename from to
```

Allows easy mass unselection of images in the sequence `sequencename` (from `from` to `to` included). See SELECT

Links: `select`

- The third includes or excludes the images selected from the list (multiple selections can be done with Ctrl or Shift) of the sequence.
- The last button can be deactivated to not show the red rectangle over registered images. It represents the framing of the reference image as computed by the registration.
- The button `Reference image` is used to select the reference image for the sequence. All sequences must have one, it will be the first image if unset or by default.

Командная строка Siril

```bash
setref sequencename image_number
```

Устанавливает опорное изображение в последовательности, указанной в первом аргументе

- Finally, the search field allows you to find the images by name.

It is also possible to sort all the images by clicking on the column headers. Thus you can sort the images by their name, their number, their X/Y offset or their FWHM. The latter is very useful to have a look at the best and worst images.
5.6 Экспорт последовательности

![Sequence export interface](image)

Рис. 3: Give a name and specify output format to export a sequence.

The **Sequence Export** tool allows you to export a sequence of images in a variety of formats. It is particularly useful if you want to export the images taking into account the registration information contained in the **seq** file, with optional cropping and normalization.

With the sequence export function, you can select a sequence to export, choose the file format and compression level for video formats. Siril's sequence export function supports a wide range of image file formats, including **FITS** (single FITS file or sequence FITS file), **TIFF**, **SER**, **AVI**, **MP4** and **WEBM** and can come in handy when building timelapse.

The button **Normalize images** allows you to normalize the images with respect to the reference image. The normalization is the same as the one done **during the stacking**, with the following settings: **Additive with scaling**, **Faster normalization** disabled.

Moreover, it is possible to play with the **image filtering** criteria to exclude or not images according to their quality. A button **Go to the stacking tab** has been added here, to easily go to the tab that exposes them.

### 5.7 Sequence information

All sequence information, registration transformation, statistics and frame selection are stored in a **.seq** file saved next to the sequence files. It is strongly recommended never to edit this file manually because Siril writes continuously inside it and one wrong character could make the reading of the sequence corrupt.

One way to clean the content of this sequence file is to go in the **Sequence** tab and click on **Clean Sequence**. The choice of what will be cleaned can be defined by clicking on the small arrow next to it.

![Clean Sequence menu](image)

Рис. 4: Menu to clean the sequence file.

Командная строка Siril
This command clears selection, registration and/or statistics data stored in `sequencename`.

You can specify to clear only registration, statistics and/or selection with `-reg`, `-stat` and `-sel` options respectively. All are cleared if no option is passed.
Astrophotography is the process of capturing images of celestial objects. It involves several steps, including preprocessing and processing, which are distinct but related.

**Preprocessing** is the initial step of working with raw astrophotographic data. It involves preparing these data for further processing. This step typically involves dark current subtraction, flat field correction, and correction of other basic problems such as removing hot and cold pixels.

**Processing** refers to the post-processing of the preprocessed data, generally after stacking. This is where the astrophotographer applies various techniques to enhance the final image and bring out details and features. These may include sharpening (deconvolution), color calibration, noise reduction, and stretching the image to increase the visibility of faint details.

In short, preprocessing sets the stage for processing by ensuring that the data is in an appropriate form and cleaned of unwanted signal, while processing is about bringing out the best in the signal to create the final image. Both steps are important in the astrophotography process, and the quality of the result depends on the skills and techniques applied at both stages.

In Siril, the main preprocessing is done following the order of the tabs in the right pane and requires the use of master files. This is a process that can be automated quite easily and the scripts provided in Siril perform this task. The image processing is processed via the dedicated menu *Image Processing*. This process is more difficult to automate because it is specific to each image and consists of an iterative work.
This section takes you through the different steps of pre-processing your images, from import into Siril to obtaining a stacked image.

The right pane contains the tabs that are useful during preprocessing. They have been designed to be used from left to right throughout the process, with some exception for the creation of masters. These tabs are also accessible via the keys F1 to F7.

Pre-processing is the step that starts with the conversion to the stacking of the images. The goal is to remove all unwanted signals and to reduce the noise present on all the subs.

### 7.1 Conversion

Siril supports the FITS 32bits format as well as the SER format in a native way. Therefore, any other file format must first be converted to these formats in order to be supported and to generate a sequence. The type of supported files is indicated in the tab and depends on how Siril was compiled.

Siril provides a conversion tab which is divided into 2 panels. The upper panel allows you to load the source files you wish to convert.

The management of these files is done from the mini toolbar:

- The first button, the + button, is the one that allows to load all the source files. It opens a dialog window allowing you to choose all the files to be converted on your computer. Only the formats supported by Siril are visible.

  **Совет:** It is possible to drag and drop files directly into the sources area. The drop zone is highlighted when the files are over it.

- The second button, the - button, allows to delete the selected files. Several files can be deleted at the same time. They are not deleted from the hard disk, but only from the conversion area.
Рис. 1: Image 1 shows the result of the conversion of a raw digital camera image. You can see visible dust, similar to dark spots. Image 2, after calibration of the images by the master darks, bias and flats shows the complete removal of these spots and a cleaner signal. Image 3 is the same after demosaicing, showing color and a very large green cast due to the higher sensitivity of the green photosites on the sensors. Finally, image 4 is the stacking output, with channel balancing.
Рис. 2: Source panel of the conversion tab.
• The last button allows you to delete all loaded files at once.

The number of loaded and selected files are reported in the status bar, to the right of the toolbar.

In the destination section it is possible to choose the name of the sequence that will be generated after the conversion of the files.

Рис. 3: Destination panel of the conversion tab.

Thus, for a sequence name basename, the converted files will be of the form

```
basenameXXXXX.[ext]
```

The extension is as defined in the preferences. The XXXX index starts by default at 00001 with the first image, however it is possible to define a different starting index. This can be useful in the case of a multi-session that shares the same master files. Three types of outputs are possible, to choose from a drop-down menu:

• FITS images
• SER sequence
• FITS sequence

These file formats are explained in the sequence section of this documentation.

Technically, when the input files are in FITS format, there is no need to convert them. However, you may want to do so so that the files are renamed to create a sequence and can be processed in Siril. In order not to fill the hard disk unnecessarily, it is then possible to choose the option Symbolic link. This option creates a symbolic link for the FITS files instead of copying them. This option is therefore only available when the output files are FITS images.

Примечание: When symbolic links are enabled, this disables compression.

Предупреждение: For Microsoft Windows, the use of symbolic links requires the activation of the developer mode in Windows.

Предупреждение: If on GNU/Linux you see the error Symbolic link Error: Function not implemented could be because you try making a symlinked sequence in a directory on a filesystem that doesn’t permit symbolic links.

When the output formats are SER, or FITS sequence, then the Multiple sequences option becomes visible. Tick this to create several sequence files instead of a single SER or FITS file for all input elements. Use this...
if input elements (sequence files such as films, SER or FITS cubes) don’t share the same image size or must not be processed together.

The last option Debayer allows the user to demosaic the images during the conversion. This option should generally not be used if the images are bias, dark and flat images, or light images intended to be pre-processed. Indeed, due to Bayer matrix consideration, the RGB result of your RAW image is an interpolated picture. In consequence pre-processing interpolated data will give wrong results. Converting RAW files of an OSC sensor gives Color Filter Array (CFA) monochrome FITS pictures. Contrary to RGB image, CFA image represent the entire sensor data with the Bayer pattern. The following image shows you a crop of a CFA image. Note that the Bayer pattern (RGGB on this example) is visible.

![Bayer pattern showed on a CFA (Color Filter Array) image.](image)

Finally, the button Convert, allows, as its name indicates, to start the conversion of files.

**Примечание:** The raw images of digital SLRs depend on the manufacturer and are generally closed source formats. Therefore the decoding of such files is a complex task that must be done by a dedicated code. For Siril, the task of converting raw files is performed by LibRaw. In fact, if a file format, usually a recent one, does not read, you have to look on the LibRaw website if it is supported. If it is not, providing them with a raw file can help the dev team to do so. However, it is also possible that the version of LibRaw embedded in the Siril package is not the most recent version. In this case, you must either wait for a new release or
7.1.1 Correspondance file

After each conversion, a file ending with _conversion.txt is created. It contains the correspondence between the input images and the images of the sequence obtained during the conversion.

Командная строка Siril

$$\text{convert basename [-debayer] [-fitseq] [-ser] [-start=index] [-out=]}$$

Converts all images in a known format into Siril's FITS images.

The argument basename is the basename of the new sequence. For FITS images, Siril will try to make a symbolic link. If not possible, files will be copied.

The flags -fitseq and -ser can be used to specify an alternative output format, other than the default FITS.

The option -debayer applies demosaicing to images. In this case no symbolic link are done.

-start=index sets the starting index number and the -out= option converts files into the directory out

Командная строка Siril

$$\text{convertraw basename [-debayer] [-fitseq] [-ser] [-start=index] [-out=]}$$

Converts DSLR RAW files into Siril's FITS images.

The argument basename is the basename of the new sequence. For FITS images, Siril will try to make a symbolic link. If not possible, files will be copied.

The flags -fitseq and -ser can be used to specify an alternative output format, other than the default FITS.

The option -debayer applies demosaicing to images. In this case no symbolic link are done.

-start=index sets the starting index number and the -out= option converts files into the directory out

7.2 Calibration

Once a sequence is loaded, images can be calibrated, registered and stacked. The calibration is an optional, yet important, step and involves bias, dark and flat frames. Calibrating a sequence in Siril can only be done with master bias, dark and flat frame, which have to be created from their sequences first.
7.2.1 Master files

![Image calibration interface](image)

Рис. 5: Masters settings of the Calibration Tab

**Bias**

Citing from *A Glossary of CCD terminology*, to explain what a bias image is:

*The bias level of a CCD frame is an artificially induced electronic offset which ensures that the Analogue-to-Digital Converter (ADC) always receives a positive signal. All CCD data has such an offset which must be removed if the data values are to be truly representative of the counts recorded per pixel.*

To use master-bias in Siril, click on the button to the right of the text entry and browse your files to select the right master. You can even use master-bias from a library as defined in the preferences.

**Совет:** The bias frame must be taken with the shutter closed and the shortest possible exposure time. Basically it corresponds to an exposure of 1/4000s with modern DSLRs.

**Synthetic bias**

Since the offset signal is very uniform on modern sensors, we recommend processing it as a constant level image. This has the advantage of saving disk space and minimizing noise in the final image. For this purpose, Siril has a feature that makes it very easy to do.

During preprocessing of your flats, instead of specifying a masterbias, you can directly type expressions in the folder selector such as:

\[=2048\]

or, if the FITS header contains the OFFSET keyword,

\[=64\times\text{OFFSET}\]

The \(=\) and \(\$\) tokens are mandatory. The level must be given in ADU (not float, even if you are working in 32-bit).

Translated into the scripting language, this is written:

```
preprocess flat -bias="=64\times\text{OFFSET}"
```

The value 2048 is here an example taken for cameras whose master-bias would have a median value of 2048. Generally, for DSLRs, the value is proportional to a root of 2. In our example, \(2048 = 2^{11}\).
Рис. 6: Example of a bias frame that was shot with a Canon EOS 1100D. Do not rely on the slightly visible bias signal, the image is auto-stretched and the differences in signal amplitudes are very exaggerated.
For more details, please refer to the tutorial on the Synthetic biases.

**Dark**

Dark frames contain the thermal noise associated with the sensor, the noise being proportional to temperature and exposure time. Hence, they should be made at approximately the same temperature as the light frames, this is the reason why we make dark frames at the end, or in the middle of the imaging session.

To use master-dark in Siril, click on the button to the right of the text entry and browse your files to select the right master. You can even use master-dark from a library as defined in the preferences.

**Совет:** Dark frames are made at the same exposure time and ISO/Gain than the subject light frames but with the shutter closed.

Рис. 7: Example of a dark frame taken with a Canon EOS 1100D with an 300s exposure and at ISO 800.

Рис. 8: An animation showing the removal of the thermal signal thanks to the dark subtraction.
Dark optimization

With the option *Optimization*, dark subtraction can be optimized so that the noise of the resulting picture (light frame minus dark frame) is minimized by applying a coefficient to the dark frame. This option is useful when dark frames have not been taken in optimal conditions.

Flat

Telescopes generally do not illuminate the detector evenly. In addition, dust on the optical surfaces and the sensor causes darker patterns in the resulting image, and the sensor itself reacts differently to the number of photons striking different photosites. To correct for these effects, each bright image must be divided by the master flat, which should be the median of the single exposures of a homogeneously and unsaturated area.

To use master-flat in Siril, click on the button to the right of the text entry and browse your files to select the right master. You can even use master-flat from a library as defined in the *preferences*.

Рис. 9: Example of a flat frame that was shot with a Canon EOS 1100D. The dust present on the optical path, and especially on the sensor, are clearly visible. The vignetting (darkening of the corners of the image) is also very visible. The defects are exaggerated by the viewing mode. Also, the *grey_flat* command was used on this image to get rid of the Bayer pattern.
**CFA equalization**

The *Equalize CFA* option equalizes the mean intensity of RGB layers in a CFA flat image. This is equivalent to apply the *grey_flat* command.

---

**Командная строка Siril**

```plaintext
grey_flat
```

Выравнивает среднюю интенсивность слоёв RGB в изображении CFA

---

**Auto evaluate normalisation value**

If the *Auto evaluate normalisation value* option is checked, Siril will auto-evaluate the normalisation value. This value is the mean of the master-flat calibrated with the master-bias. Otherwise, the value indicated in the text box will be taken into account.

---

**7.2.2 Calibration of light images**

The calibration of the light images consists in applying the master bias, dark and flat to the astronomical images in order to remove the unwanted signal.

**Предупреждение:** In no case does the calibration reduce the noise of the images. On the contrary, it increases it. This is why it is important to take as many calibration images as possible, such as darks, in order to minimize the amount of noise in the images.

---

**Исправить артефакты автофокуса в X-Trans**

This *Fix X-Trans AF artifact* option helps to fix the Fujifilm X-Trans Auto Focus pixels. Indeed, because of the phase detection auto focus system, the photosites used for auto focus get a little less light than the surrounding photosites. The camera compensates for this and increases the values from these specific photosites giving a visible square in the middle of the dark/bias frames. This option has no effect on Bayer pattern. The option is only enabled if a master-bias or master-dark is loaded and used.

---

**Косметическая коррекция**

Cosmetic correction is the technique used to correct defective pixels in images. Indeed, any camera sensor has photosites that do not react correctly to photon reception. This is visible in the image with pixels with values very different from those of their nearest neighbors. These pixels are called *hot pixels*, if the value is much higher, or *cold pixels* when it is much lower. Siril offers two algorithms to correct these defective pixels if the option *Enable Cosmectic Correction* is checked.
Использовать темновой мастер-кадр

This method requires the presence of a master-dark. Siril will search for pixels whose deviation from the median exceeds $x$ times the standard deviation $\sigma$. This value is adjustable for both hot and cold pixels.

It is possible to estimate the number of pixels that will be corrected in the calibrated image by pressing the *Estimate* button. If the corrected pixel value is displayed in red, it means that this number exceeds 1% of the total number of pixels in the image. In this case you should increase the value of the coefficient, or uncheck the corresponding correction. If the images are from a color sensor then it is necessary to check the *CFA* option.
Using Bad Pixel Map

The other method uses a file that contains the coordinates of the defective pixels. This file is a simple text file and can initially be created with the command `find_hot`.

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

Командная строка Siril

`find_hot filename cold_sigma hot_sigma`

Saves a list file `filename` (text format) in the working directory which contains the coordinates of the pixels which have an intensity `hot_sigma` times higher and `cold_sigma` lower than standard deviation. We generally use this command on a master-dark file

```
Lines P x y type will fix the pixel at coordinates (x, y) type is an optional character (C or H) specifying to Siril if the current pixel is cold or hot. This line is created by the command FIND_HOT but you also can add some lines manually:

Lines C x 0 type will fix the bad column at coordinates x.
Lines L y 0 type will fix the bad line at coordinates y.
```

This file, which can be edited by hand, is to be loaded as a Bad Pixel Map.

Finally, if the images are from a color sensor then it is necessary to check the CFA option.

Итоговая последовательность

This section groups the options that can be applied to the output.

- The Output Prefix entry box adds a prefix to the output images, to easily identify them. By default, the prefix is `pp_`, which means pre-processed.
- The drop-down list defines the type of destination sequence.
  - FITS images: one FITS file per image.
  - SER sequence: one SER file for the whole sequence (limited to 16 bits per channel).
  - Sequence FITS: one FITS file for the entire sequence.
- Last option, Debayer before saving. Check this option if you want to apply a demosaicing algorithm to your frames right after they were calibrated. By doing this, you skip one manual step which can take some time.
Command lines

Командная строка Siril

```
preprocess sequencename [-bias=filename] [-dark=filename] [-flat=filename] [-cc=dark␣
˓→opt] [-all] [-prefix=] [-fitseq]
```

Calibrate the sequence `sequencename` using bias, dark and flat given in argument.

For bias, a uniform level can be specified instead of an image, by entering a quoted expression starting with an = sign, such as `-bias="=256"` or `-bias="=64*$OFFSET"`.

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with `-cc=` option.

You can use `-cc=dark` to detect hot and cold pixels from the masterdark (a masterdark must be given with the `-dark=` option), optionally followed by `siglo` and `sighi` for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied.

Alternatively, you can use `-cc=bpm` followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with a `find_hot` command on a masterdark.

It is possible to specify if images are CFA for cosmetic correction purposes with the option `-cfa` and also to demosaic images at the end of the process with `-debayer`.

The `-fix_xtrans` option is dedicated to X-Trans files by applying a correction on darks and biases to remove an ugly square pattern.

The `-equalize_cfa` option equalizes the mean intensity of RGB layers of the CFA flat master.

It is also possible to optimize the dark subtraction with `-opt`.

By default, frames marked as excluded will not be processed. The argument `-all` can be used to force processing of all frames even if marked as excluded.

The output sequence name starts with the prefix "pp_" unless otherwise specified with option `-prefix=`.

If `-fitseq` is provided, the output sequence will be a FITS sequence (single file).

Командная строка Siril

```
```

Calibrate the image `imagename` using bias, dark and flat given in argument.

For bias, a uniform level can be specified instead of an image, by entering a quoted expression starting with an = sign, such as `-bias="=256"` or `-bias="=64*$OFFSET"`.

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with `-cc=` option.
You can use `-cc=dark` to detect hot and cold pixels from the masterdark (a masterdark must be given with the `-dark=` option), optionally followed by `siglo` and `sighi` for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied.

Alternatively, you can use `-cc=bpm` followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with a `find_hot` command on a masterdark.

It is possible to specify if images are CFA for cosmetic correction purposes with the option `-cfa` and also to demosaic images at the end of the process with `-debayer`.

The `-fix_xtrans` option is dedicated to X-Trans files by applying a correction on darks and biases to remove an ugly square pattern.

The `-equalize_cfa` option equalizes the mean intensity of RGB layers of the CFA flat master.

It is also possible to optimize the dark subtraction with `-opt`.

Note that the command-line parser will not reject the argument `-pex`, which is used in preprocessing sequences, but it will be ignored.

The output filename starts with the prefix "pp_" unless otherwise specified with option `-prefix=`.

---

### 7.2.3 Understanding how the flats correct the lights

The point of this section is to give a bit more insight in how the different levels play a role in the correction of the lights by the flats.

We will disregard here any considerations about noise (again, noise does not vanish with masters subtraction or division, it decreases by averaging over many realizations of the same random process). We also disregard particular spatial patterns such as ampglow or dust.

If we try to quantify the intensity of background pixels in the different frames we have, we can write the following expressions:

$$L = a - b \times \left( x - \frac{W}{2} \right)^2 + d_{\text{rate}} \times t_{\text{lights}} + o$$  \hspace{1cm} (7.1)$$

$$D = d_{\text{rate}} \times t_{\text{lights}}$$

$$F = K \left( a - b \times \left( x - \frac{W}{2} \right)^2 \right) + o$$  \hspace{1cm} (7.3)$$

with, $L$ for Lights, $D$ for Darks, $F$ for Flats and $O$ for Bias.

For the lights $L$, the first part is a spatial illumination component, i.e., $a - b(x - \frac{W}{2})^2$. We have chosen here a quadratic variation with a maximum value $a$ in the middle of the frame of width $W$, even about the center of the sensor. This is not the exact spatial shape of vignetting but it is a good enough approximation to understand how it works. In addition to this spatial illumination term, there is a term varying with exposure time which is usually named dark current ($d_{\text{rate}} \times t_{\text{lights}}$) but which does not depend on the position of the pixel on the sensor. And finally there is a pedestal, i.e. the offset. This offset is present in any frame which is shot, so that we find it in all the expressions.

The darks $D$ not being illuminated, they only bear the dark current term, with same intensity as lights as they are shot for the same amount of time, and the offset term.

The flats $F$ also have a spatial term, proportional to the term found in the lights. The factor $K$, larger than 1, simply shows that their intensity is larger. To write this, we only need to assume that the pixels
respond linearly to the number of photons they gather, which is sensible. We could also have written a dark current term, proportional to the exposure time of flats. But unless this time is significant, we can assume it is negligible. If it is not the case, then it means you need to shoot dark flats, or at least to assess their level.

And finally the offsets $O$ only measure the offset level.

To visualize these levels, we have plotted here-below these expressions as curves wrt. position on a frame and we encourage you to do the same and to play around with the inputs.

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

$L$, $D$ and $O$ values in ADU are given on the left scale while $F$ are on the scale reported to the right.

Now what does calibrating your lights mean? When you calibrate your lights, you perform the following operation:

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

The term $F - O$ is a flat from which you have subtracted the offset level (whether it is a masterbias or simply a level). This is the operation performed prior to stacking your masterflat. And the term $L - D$ represents a
light from which you have subtracted the dark current level and the offset, *i.e.* a masterdark. If you replace with the expressions shown above, you end up with the following:

\[ L_c = \frac{1}{K}. \]

No spatial variation term is left, you have flat-fielded your lights! Getting a sensible value in ADU (and not \(1/K\)) is what Siril does when you check *Auto evaluate normalisation value* in the *Calibration* tab.

And you can try with any other combination, no other will get rid of spatial variations.

Just to illustrate this, We have plotted below the result of different combinations. To put everything on the same scale, all the results are normalized to have the same intensity of 1 in the middle of the frame. The following tests are shown:

- \(L - D\) : you have just shot darks.
- \(L/F\) : you have just shot flats.
- \(L/(F - O)\) : you have shot flats and corrected them by an offset (either a master or a synthetic one).
- \((L - O)/(F - O)\) : you have just flats corrected by offset. But you have subtracted the offset from your lights as well.
- \((L - D)/F\) : you have shot flats and darks but no offsets.
- \((L - D)/(F - O)\) : you have done everything by the book.

![Intensity vs position on frame.

Interestingly, you can notice that:
• $L − D$ shows obviously no correction for vignetting.

• Both $L/F$ and $L/(F − O)$ show overcorrection or inverse vignetting.

• Getting very close to the optimal result, $(L − D)/F$ and $(L − O)/(F − O)$ shows a field almost flat. This, of course, will depend how much your sensor has dark current and how much vignetting your optical train shows.

• The reference calibration gives a flat field.

The conclusions that you can draw from the above are:

• You are better off correcting your lights with offset (masterbias or simply a level) if you have not shot darks.

• Even better, if you don’t have time to shoot a series of darks, it is probably worth shooting at least one dark, measure its median, and subtract this (synthetic) dark from your lights. It will of course not correct for ampglow or enable hot pixel correction, but your lights will at least be flat!

**Now what about dust...?**

In order for your flats to also correct for these nasty spots, the sad news is you also need to get all the calibration frames in the equation. We have added a small local ADU deficit in the lights and flats to illustrate this effect.

As you can see, only the combination $(L − D)/(F − O)$ can get rid of it.
To further illustrate the equations and curves above, nothing is better than a real-life example. All pictures below are shown courtesy of G. Attard.

Рис. 11: $L - D$

### 7.3 Registration

Registration is basically the process of aligning the images from a sequence to be able to process them afterwards. All the processes described hereafter calculate the transformation to be applied to each image in order to be aligned with the reference image of the sequence.

Siril's strength lies in the wide variety of recording algorithms offered. Each method is explained below. Pressing the Go register button starts the registration of the sequence.

It is possible to choose the **registration channel**. Green is the default for color images, Luminance for monochrome. The (*) sign appearing after the channel's name means that registration data is already available for this layer. When processing images, registration data is taken from the default layer if available (for RGB images: Green, else fallback to Blue then Red).
Рис. 12: $L/F$
Рис. 13: $L/(F - O)$
Рис. 14: \( (L - O)/(F - O) \)
Рис. 15: \( (L - D)/(F - O) \)
7.3.1 Theory

Registration process

What we call Registration is in fact a three-step process:

1. Detect features to be matched in all the images
2. Compute the transformations between each image and the reference image
3. Apply the computed transformation to each image to obtain new images

Depending on the registration method selected, the 3 steps occur (or not) into a single process. Siril uses the most sensible defaults (choosing or not to apply the computed transformation) depending on the registration method selected, but understanding the internal machinery may help you to change this behavior to better suit your needs.

Algorithms

The table here below details the different algorithms used for the first 2 steps (detection and transformation calculation).

<table>
<thead>
<tr>
<th>Registration method</th>
<th>Feature detection</th>
<th>Transformation calculation</th>
<th>Сдвиг</th>
<th>Евклидово</th>
<th>Сходство</th>
<th>Аффинное</th>
<th>Гомография</th>
</tr>
</thead>
<tbody>
<tr>
<td>Все цвета 2 pass</td>
<td>Динамическая PSF</td>
<td>Triangles matching + RANSAC</td>
<td>subpixel</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>1-2-3 звезды</td>
<td>PSF minimization</td>
<td>Singular Decomposition (2-3 stars)</td>
<td>Value subpixel (1 star) (2-3 stars)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image Pattern alignment</td>
<td>cross correlation</td>
<td>on selection box</td>
<td>pixel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KOMBAT</td>
<td>Max of convolution in spatial domain on selection box</td>
<td>pixel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comet</td>
<td>PSF minimization in selection box</td>
<td>Shifts from velocity vector using timestamps</td>
<td>subpixel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manual</td>
<td>Your eyes</td>
<td>Your hand</td>
<td>pixel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It is also important to keep in mind how the registered sequence is fed into the stacking process that is generally used right after registration:

- If the transformation consists only of pixel-wise shifts, the stacking algorithm can use these shifts on-the-fly when reading the images. It means you do not need to generate "registered images". This saves storage space and skips interpolation. It is, of course, at the expense of less accurate registration (i.e. subpixel accuracy) but is generally used on planetary/lucky imaging images where sampling is small. This can also be applied with a registration method which computes subpixel shifts. During the stacking process, the shifts will be rounded off to pixel precision. In any other case, meaning the stacking is fed with a sequence where the registration has computed transformations more complex than just shifts but the registered images have not been saved, Siril will emit a warning inviting you to export the registered images before proceeding to stacking.

- In all other cases, once the transformations have been computed, the transformed images need to be saved before proceeding to stacking, generally named with \( r \) prefix.
**Images transformations**

Siril uses linear transformations, with different degrees-of-freedoms, to map an image to the reference image:

- **Shift** is a 2 degree-of-freedom (x/y shifts) rigid mapping, well-suited for images with no distortion, no scaling and no field rotation. It needs only 1 pair of stars (or feature) to be matched to define the transformation.

- **Euclidean** is a 3 degree-of-freedom (x/y shifts + one rotation) rigid mapping, for images with no distortion, no scaling. It needs at least 2 pairs of stars to be matched to define the transformation.

- **Similarity** is a 4 degree-of-freedom (one scale, one rotation and x/y shifts) more rigid mapping than homography, well-suited for images with no distortion. It needs at least 2 pairs of stars to be matched to define the transformation.

- **Affine** is a 6 degree-of-freedom (two scales, one shear, one rotation and x/y shifts) more rigid mapping than homography, well-suited for images with little distortion. It needs at least 3 pairs of stars to be matched to define the transformation.

- **Homography** is the default transformation which uses an 8-degree-of-freedom transform to warp the images onto the reference frame. This is well-suited for the general case and strongly recommended for wide-field images. It needs at least 4 pairs of stars to be matched to define the transformation.
Reference image

This is the image which is used as a common reference to compute the transformations that send all the images of the sequence onto this particular one.

If not set manually, the reference image is chosen with the following criteria:

- if the sequence has already been registered, it is the best image, in term of lowest FWHM or highest quality depending on the type of registration
- Otherwise, it is the first image of the sequence that is not excluded.

To specify an image as the reference, you can:

- Open the Frame selector, select the image to be set as the new reference and click the button Reference Image.
- Use the command setref. For instance, if you want to set image #10 as the reference:

  setref 10

Командная строка Siril

setref sequencename image_number

Устанавливает опорное изображение в последовательности, указанной в первом аргументе

Рис. 16: The frame list dialog. You can browse all images in the sequence.
During stacking, the reference image is used as the normalization reference as well, if normalization is activated.

### 7.3.2 Registration methods

#### Global registration

This is probably the more powerful and accurate algorithm to align deep-sky images.

The global matching is based on triangle similarity method for automatically identify common stars in each image [Valdes1995]. Our implementation is based upon the program `match` from Michael Richmond. Then, RANSAC [Fischler1981] algorithm is used on the star lists to further reject outliers and determine the projection matrix. The robustness of the algorithm depends on the ability to detect the stars while avoiding false detections. Siril has a very elaborate star detection algorithm that avoids as much as possible to select objects that are not stars in the fastest possible time. The detection of the brightest stars is usually the most important. However, if there is a need to detect fainter stars, then the *Dynamic PSF* window can be used to adjust the detection parameters.

![Automatic detection of stars in a single frame](image)

Рис. 17: Automatic detection of stars in a single frame

There are few options associated with this alignment method because it is fairly automatic. The *Transformation* dropdown menu allows to choose between different transformations.

7.3. Registration
Предупреждение: The initial star matching uses triangle similarity algorithm, in consequence the minimum of star pairs must be at least of 3 for Shift, Similarity and Affine and of 4 for Homography.

Other options are:

- The Minimum Star Pairs button sets the minimum number of star pairs a given frame can have in relation with the reference frame. If a given light frame has less star pairs, it will not be registered. To the right of this option is a button that opens the PSF Dynamique tool.

- The option Maximum Stars Fitted defines the maximum number of stars to be searched for in each frame (default 2000). The larger this value, the more stars will potentially be detected, resulting in a longer detection but more accurate registration.

- Finally, the last option, Match stars in selection, if you want to perform the Global Star Alignment algorithm within the selected area in the reference image. If no selection are done, this option is ignored.

Командная строка Siril

```
```

Finds and optionally performs geometric transforms on images of the sequence given in argument so that they may be superimposed on the reference image. Using stars for registration, this algorithm only works with deep sky images. Star detection options can be changed using SETFINDSTAR or the Dynamic PSF dialog. The detection is done on the green layer for colour images, unless specified by the -layer= option with an argument ranging from 0 to 2 for red to blue.

The -2pass and -noout options will only compute the transforms but not generate the transformed images, -2pass adds a preliminary pass to the algorithm to find a good reference image before computing the transforms. -nolist disables saving the star lists to disk.
The option `-transf=` specifies the use of either *shift*, *similarity*, *affine* or *homography* (default) transformations respectively.

The option `-drizzle` activates the sub-pixel stacking by up-scaling by 2 the generated images.

The option `-minpairs=` will specify the minimum number of star pairs a frame must have with the reference frame, otherwise the frame will be dropped and excluded from the sequence.

The option `-maxstars=` will specify the maximum number of star to find within each frame (must be between 100 and 2000). With more stars, a more accurate registration can be computed, but will take more time to run.

The pixel interpolation method can be specified with the `-interp=` argument followed by one of the methods in the list `no[ne], ne[arest], cu[bic], la[nczos4], li[near], ar[ea]`. If `none` is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the `-noclamp` argument.

All images of the sequence will be registered unless the option `-selected` is passed, in that case the excluded images will not be processed.

If created, the output sequence name starts with the prefix "r_" unless otherwise specified with `-prefix=` option.

Links: *setfindstar*, *psf*

### 2pass registration

The global star alignment is done in two passes, allowing the reference frame to be chosen from detected star information instead of automatically choosing the first frame of the sequence. The proposed options are similar to the *Global Registration* algorithm but this method does not create any sequences and all alignment information is saved in the `seq` file.

During star detection, Siril sets a maximum of 2000 stars to be found (this can also be changed with the appropriate option). In case more than one image has reached the maximum star limits, the star lists of all images are screened again. A new minimum detection threshold is defined to be able sort the images by both number of stars detected and FWHM.

Unless specified otherwise, the star lists of all images are saved when using this method, the `.fit(s)` extension being replaced by `.lst`. This allows to re-run the 2pass algorithm very quickly with different parameters, say different transformation. In case the star detection have been modified, the process detects these changes and re-run the analysis as required.

This registration must generally followed by *Apply Existing Registration* in order to apply the transformation and build a new sequence, unless you have chosen to compute *Shift*.

These lines perform a 2pass registration on a sequence named `pp_light` and applies it. The output is a sequence `pp_light`.

```bash
# Align lights in 2 passes
register pp_light -2pass
seqapplyreg pp_light
```
These lines perform a 2pass registration on a sequence named *colors* and applies it while cropping the output images to the minimum common area. The output is a sequence *pp_colors*. This can be useful before compositing mono images (the areas which are not common to all images are cropped).

```
# Align layers in 2 passes and crop away borders
register colors -2pass
seqapplyreg colors -framing=min
```

### 1-2-3 stars registration

When the images contain few stars, for example in the case of DSO Lucky Imaging images where the frame exposure is less than one second. It is possible that the global registration algorithm fails, even if you change the detection parameters in the *Dynamic PSF* window. It may then be interesting to make a manual detection of the stars you want to align. This is the interest of the 1, 2 or 3 star registration algorithm.

The principle of this method is to draw a selection area around a star and click on the *Pick 1st star* button, then so on.

- If only one star is selected, only the translation between the images will be calculated. Therefore the *Shift only* button is automatically selected. The shift values are then stored in the *seq* file.

- If two or three stars are selected, then the rotation can be calculated and applied to create a new sequence. However, if the *Shift only* option is selected, which is not mandatory, only the shifts will be calculated.

The option *Follow star movement* use the position of the star(s) found in the previous image as new centre for the current image registration. This allows the selection area to be smaller, registration faster, and accounts for drift or images with a large number of stars.

**Предупреждение:** Enabling this option requires the registration to not be parallelized, it will run on one CPU core only.
Image Pattern alignment (planetary-full disk)

This is a simple registration by translation method using cross correlation in the spatial domain. This method is fast and is used to register planetary movies, in which contrasted information can be seen on large areas of the image. It can also be used for some deep-sky images registration. Nevertheless keep in mind that it is a single point alignment method, which makes it poorly suited for high definition planetary alignment. But, it does effectively anchor the images to stabilize the sequence. Simply draw a selection around the object (the planet for example) and make sure that its movement during the sequence is contained within the selection. Only the translation can be calculated with this method.

KOMBAT

This method comes from the OpenCV library, a library widely used in Siril. They explain:

It simply slides the template image over the input image (as in 2D convolution) and compares the template and patch of input image under the template image. Several comparison methods are implemented in OpenCV. (You can check docs for more details). It returns a grayscale image, where each pixel denotes how much does the neighbourhood of that pixel match with template.

In practice, simply draw a selection around the object (the planet for example) and make sure that its movement during the sequence is contained within the selection. Only the translation can be calculated with this method.
Comet/Asteroid registration

The cometary registration tool works in a very simple way, in two steps.

1. With the frame selector, select the first image of the sequence, surround the comet nucleus, then click on the button *Pick object in #1*.

2. Then select the last image of the sequence, surround the nucleus of the comet, then click on the button *Pick object in #2*.

The comet velocity $\Delta x$ and $\Delta y$ is computed in pixel per hour if everything is ok.

Предупреждение: The alignment of the comet must be done on images whose stars have been previously aligned. Either via a new sequence, with the global alignment, or by having saved the registration information in the seq file. In this last case, the option *Accumulate reg. data* (explained below) makes sense.

Примечание: To fully function, the images must have a timestamp. Only FITS, SER and TIFF images are compatible with this feature.

Manual Registration

This last method of registration is very particular, which explains its separate position, and allows to align images manually. Of course, only the translation between images is allowed.

The first thing to do is to define two previews in the image. Clicking on the button *Set first preview* will initialize the first preview. You then need to click on an area of the image, ideally a star in the vicinity of an edge of the image to set the preview area. A click on the second button *Set second preview* allows to do the same on a second point.

It is very important to have a reference image already set with the *Frame selector*. By default, it is the first image. The user is free to choose the one he wants. It will be used as a reference layer, seen by transparency,
7.3. Registration
to align the images manually with the numerical buttons. Then, browse the image one by one to apply the same method to the whole sequence.

Рис. 18: The Y-shift is too large, same stars on different frames do not overlap.

**Apply Existing registration**

This is not an algorithm but rather a commodity to apply previously computed registration data stored in the sequence file. The interpolation method and simplified drizzle can be selected in the *Output Registration* section. You can also use image filtering to avoid saving unnecessary images, as in stacking *Image rejection*.

Four framing methods are available:

- **current**: uses the current reference image. This is the default behavior.
Рис. 19: X- and Y-shift look fine. The current image is aligned to the reference one.
• **maximum** (bounding box) adds a black border around each image as required so that no part of the image is cropped when registered.

• **minimum** (common area) crops each image to the area it has in common with all images of the sequence.

• **center of gravity** determines the best framing position as the center of gravity (cog) of all the images.

Командная строка Siril

```
seqapplyreg sequencename [-drizzle] [-interp] [-noclamp] [-layer] [-framing] [-prefix] [-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]]
```
Applies geometric transforms on images of the sequence given in argument so that they may be superimposed on the reference image, using registration data previously computed.

The output sequence name starts with the prefix "r_" unless otherwise specified with -prefix= option.

The option -drizzle activates up-scaling by 2 the images created in the transformed sequence.

The pixel interpolation method can be specified with the -interp= argument followed by one of the methods in the list no[ne], ne[arest], cu[bic], la[nzos4], li[near], ar[ea]). If none is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the -noclamp argument.

The registration is done on the first layer for which data exists for RGB images unless specified by -layer= option (0, 1 or 2 for R, G and B respectively).

Automatic framing of the output sequence can be specified using -framing= keyword followed by one of the methods in the list { current | min | max | cog }:
- framing=max (bounding box) adds a black border around each image as required so that no part of the image is cropped when registered.
- framing=min (common area) crops each image to the area it has in common with all images of the sequence.
- framing=cog determines the best framing position as the center of gravity (cog) of all the images.

Filtering out images:
Images to be registered can be selected based on some filters, like those selected or with best FWHM, with some of the -filter-* options.

With filtering being some of these in no particular order or number:

```text
[-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-included]
```

Best images from the sequence can be stacked by using the filtering arguments. Each of these arguments can remove bad images based on a property their name contains, taken from the registration data, with either of the three types of argument values:
- 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.

It is also possible to use manually selected images, either previously from the GUI or with the select or unselect commands, using the -filter-included argument.
7.3.3 Output Registration

This frame contains all the output elements for the sequence.

- The button *Simplified Drizzle x2* activates the simplified drizzle algorithm for the processing of this sequence. An up-scale (x2) will be applied to the registered frame or during stacking depending on which registration is chosen, that will result in higher resolution images. This option is adapted for under-sampled images, *i.e.* when the telescope focal length is too short for the pixel size. One may consider that the system is under-sampled when FWHM is smaller than 2 pixels. The correct name of this method should be super-resolution stacking, but for a more convenient understanding we called it *Simplified Drizzle x2*.

- *Warning:* The counterpart of this technic is that the amount of memory and disk space needed to create and process drizzled images is multiplied by the square of the Drizzle factor.

- When button *Save transformation in seq file only* is checked, the transformed images are not saved as a newly registered sequence. In both cases, the transformation matrices are saved to the sequence file. The registration data can then be inspected and some images unselected, prior to applying the transformations using the Apply Existing Registration method. This option is automatically checked for alignment method that produce *shift only* registration data. If this option is unchecked, then it is possible to define a prefix for the new sequence that will be created. By default it is $r_\_$. 

- If a new sequence is created, with the application of a complete transformation, then the pixels of the resulting images are interpolated by an algorithm that is left to the user’s choice. There are 5 possible interpolation algorithms, plus a *None* option:
  - Ближайший сосед
  - Билинейная
  - Бикубическая
  - Отношение площади пикселя
  - Ланцош-4
The most efficient interpolation methods are generally bicubic and Lanczoz (used by default). However, they usually require the Clamping interpolation option to be enabled to avoid ring artifacts around the stars. But the latter may be useless in some cases. We recommend you to test with your images.

The special case of None is reserved for the case of global registration and Apply Existing registration. If you want to export or save a sequence that contains only translation, without using interpolation (so as not to modify the pixel values), you should select None.

- Last option Accumulate reg. data, must be checked if you want the new registration data to be added to the previous one. This option is useful when sequence has previously been aligned using a method that does not build a new sequence, but it should be unchecked when the comet/asteroid algorithm is applied several times.

7.3.4 Литература

7.4 Укладка

The final preprocessing step to do with Siril is to stack the images. Image stacking is a technique used in astrophotography to increase the quality and detail of an image by combining multiple photographs into a single, composite image. The process involves taking multiple images of the same object and then align and average the frames together to reduce the noise and increase the signal-to-noise ratio. This results in a final image that has less noise, greater detail and greater dynamic range than a single exposure.

7.4.1 Способы укладки

**Sum stacking**

This is the simplest algorithm: each pixel in the stack is summed. The increase in signal-to-noise ratio (SNR) is proportional to $\sqrt{N}$, where $N$ is the number of images. Because of the lack of normalisation and rejection, this method should only be used for planetary processing.

For 8 or 16 bit per channel input images, the sum is done in a 64 bit integer before being normalized to the maximum pixel value and saved as a 16 bit unsigned integer or 32 bit floating point image.

This stacking method should be used for 8-bit input images because it will increase the dynamic of the images while stacking them, making features discernable. Stacking with an mean or median method such a sequence would only decrease the noise but not improve the dynamic of the image, the result would still be 8 bits deep.

**Average Stacking With Rejection**

This method of stacking computes a mean of the pixels in a stack after having excluded deviant pixels and an optional normalisation of the images against the reference image. As for sum stacking, the improvement in SNR is proportional to $\sqrt{N}$. There are several ways to normalize the images and several ways to detect and replace or exclude deviant pixels, explained below.

*Предупреждение:* Some operating systems limit the number of images that can be opened at the same time, which is required for median or mean stacking methods. For Windows, the limit is 2048 images. If you have a lot of images, you should use another type of sequence, described [here](#).
Rejection methods

- **Percentile Clipping**: This is a one step rejection algorithm ideal for small sets of data (up to 6 images).
- **Sigma Clipping**: This is an iterative algorithm which will reject pixels whose distance from median will be farthest than two given values in sigma units ($\sigma_{low}, \sigma_{high}$).
- **MAD Clipping**: This is an iterative algorithm working as Sigma Clipping except that the estimator used is the Median Absolute Deviation (MAD). This is generally used for noisy infrared image processing.
- **Median Sigma Clipping**: This is the same algorithm as Sigma Clipping except than the rejected pixels are replaced by the median value of the stack.
- **Winsorized Sigma Clipping**: This is very similar to Sigma Clipping method, except it is supposed to be more robust for outliers detection, see Huber's work [Peter2009].
- **Generalized Extreme Studentized Deviate Test** [Rosner1983]: This is a generalization of Grubbs Test that is used to detect one or more outliers in a univariate data set that follows an approximately normal distribution. This algorithm shows excellent performances with large dataset of more 50 images.
- **Linear Fit Clipping** [ConejeroPI]: It fits the best straight line ($y = ax + b$) of the pixel stack and rejects outliers. This algorithm performs very well with large stacks and images containing sky gradients with differing spatial distributions and orientations.

Rejection maps

The option *Create rejection maps* computes and creates rejection maps during stacking. These are images showing how many images were rejected for each pixel of the result image, divided by the number of stacked images. If *Merge L+H* is checked, Siril creates only one rejection map that will be the sum of the low and high maps.

Images filtering/weighting

The weighting allows to put a statistical weight on each image. In this way, the images considered to be the best will contribute more than those considered to be the worst. Four methods of weighting are available:

- **Number of stars** weights individual frames based on number of stars computed during registration step.
- **Weighted FWHM** weights individual frames based on wFWHM computed during registration step.
- **Noise** weights individual frames based on background noise values.
- **Number of images** weights individual frames based on their integration time.

**Медиана**

This method is mostly used for dark/flat/bias stacking. The median value of the pixels in the stack is computed for each pixel.

The increase in SNR is proportional to $0.8\sqrt{N}$ and is therefore worse than stacking by average which is generally preferred.

**Pixel Maximum stacking**

This algorithm is mainly used to construct long exposure star-trails images. Pixels of the image are replaced by pixels at the same coordinates if intensity is greater.

**Pixel Minimum stacking**

This algorithm is mainly used for cropping sequence by removing black borders. Pixels of the image are replaced by pixels at the same coordinates if intensity is lower.

### 7.4.2 Input normalisation methods

Normalisation will adjust the levels of each image against the reference image. This is particularly useful for mean stacking with rejection, because rejecting pixels if the images show differences of levels is not very useful. They can be caused by light nebulosity, light gradient caused by the moon or city lights, sensor temperature variation and so on.

This tends to improve the signal-to-noise ratio and therefore this is the option used by default with the additive normalisation.

If one of these 5 items is selected, a normalisation process will be applied to all input images before stacking.

- Normalisation matches the mean background of all input images, then, the normalisation is processed by multiplication or addition. Keep in mind that both processes generally lead to similar results but multiplicative normalisation is preferred for image which will be used for multiplication or division as flat-field.

- Scale matches dispersion by weighting all input images. This tends to improve the signal-to-noise ratio and therefore this is the option used by default with the additive normalisation.

**Примечание:** The bias and dark masters should not be processed with normalisation. However, multiplicative normalisation must be used with flat-field frames.
Keep in mind that both processes generally lead to similar results but multiplicative normalisation is preferred for image which will be used for multiplication or division as flat field.

Since the normalisation calculation step is usually a long one, as it requires determining all the statistics of the image, the results are stored in the `seq` file. This way, if the user wants to do another stacking by changing the rejection parameters, it will be executed more quickly. The `Recompute` option allows to force the recalculation of the normalisation.

By default, Siril uses IKSS estimators of location and scale to compute normalisation. For long sequences, computing these estimators can be quite intensive. For such cases, you can opt in for faster estimators (based on median and median absolute deviation) with the option `Faster normalisation`. While less resistant to outliers in each image, they can still give a satisfactory result when compared to no normalisation at all.

### 7.4.3 Image rejection

It is also possible to reject a certain number of images in order to select only the best ones. This can be very useful for Lucky DSO techniques where the number of images in a sequence is very high. One can choose between % and k-$\sigma$ to either retain a given percentage of images or to calculate the allowable threshold using k-$\sigma$ clipping.

Several criteria are available:

- **all**: all images of the sequence are used in the stack.
- **selected**: only use image that have not been unselected from the sequence.
- **FWHM**: images with best computed FWHM (star-based registration only).
- **weighted FWHM**: this is an improvement of a simple FWHM. It allows to exclude much more spurious images by using the number of stars detected compared to the reference image (star-based registration only).
- **roundness**: images with best star roundness (star-based registration only).
- **background**: images with lowest background values (star-based registration only).
- **nb stars**: images with best number of stars detected (star-based registration only).
- **quality**: images with best quality (planetary DFT or Kombat registrations).
7.4.4 Stacking result

- If *Output Normalisation* is checked, the final image will be normalized in the [0, 1] range if you work in 32-bit format precision, or in [0, 65535] otherwise.

| Предупреждение: | This option should not be checked for master stacking. |

- If *RGB equalization* is checked, the channels in the final image will be equalized (color images only).
- The stacking result is saved under the name given in the text field. It is possible to use *path parsing* to build the filename. A click on the *overwrite* button allows the new file created to overwrite the old one if it exists. If the latter is not checked but an image with the same name already exists, then no new file is created.

7.4.5 Литература
This section takes you through the different processing steps of your images. The drop-down menu is accessible from the header bar using the *Image Processing* button. The tools are grouped in the menu, and in this documentation too, by theme.

### 8.1 Image stretching

Image are stored as pixel values that come from the camera following a quasi-linear law, meaning that for areas of the sky that show no visible feature, the pixel value will be close to zero, but for bright objects like stars it will be close to a maximum value depending on exposure and gain. In between, if a nebula has a surface magnitude half of a star, it will have pixel values half of those of the star and so on. This is what we call the linear pixel mode.

The human eye doesn't quite see photons like that. It amplifies dark areas, so that an object maybe a tenth as bright as another would look half as bright. For astronomy images, we usually display images with a similar pixel value scaling (see display modes from the GUI).

But it is only a display trick, using a screen transfer function, to render the pixel values of the untouched image to better looking images.

Image stretching is about doing something similar but by modifying the pixel values of images instead of just altering their rendering. Siril has three main tools to achieve this.
Рис. 1: Image processing menu
8.1.1 Asinh transformation

The asinh, or inverse hyperbolic sine, transformation will modify image pixel values in a way similar to what can be seen with the asinh display pixel scaling function, which is parametrized by the low and high values cut-off cursors. Here the parameters are the stretch factor and the black point value.

![Asinh Transformation Dialog Box](image)

Рис. 2: Dialog box of Asinh Transformation

For monochrome images, pixel values are modified using the following function:

\[
\text{pixel} = \frac{(\text{original} - \text{blackpoint}) \times \text{asinh(original \times stretch)}}{\text{original} \times \text{asinh(stretch)}}
\]

For color images, the function becomes:

\[
\text{pixel} = \frac{(\text{original} - \text{blackpoint}) \times \text{asinh(rgb\_original \times stretch)}}{\text{rgb\_original} \times \text{asinh(stretch)}}
\]

where rgb_original is computed using the pixel values of the three channels.

Примечание: As rgb_original is an average of the 3 channels, one or two channel values will be greater than rgb_original and can therefore clip. This can cause color artefacts when bright, strongly-colored regions are stretched. In order to avoid this problem the RGB blend clipping algorithm is used. This was devised by the same authors as the Generalised Hyperbolic Stretch transforms. The \((r, g, b)\) values are stretched first based on the luminance value rgb_original to give \((r', g', b')\). Then the original \((r, g, b)\) values are independently stretched to give \((r'', g'', b'')\). Finally the largest value of \(k\) is identified such that

\[
k \ast r' + (1 - k) \ast r''1;
\]

\[
k \ast g' + (1 - k) \ast g''1;
\]

и

\[
k \ast b' + (1 - k) \ast b''1
\]

Then the transformed values are calculated as
(k * r' + (1 - k) * r'”, k * g' + (1 - k) * g’”, k * b' + (1 - k) * b”)

This RGB blend clipping algorithm is also used for the Generalised Hyperbolic Stretch transforms described below.

When the Use RGB working space option is not ticked, rgb_original is the mean between the three pixel values; when it is set, ponderation changes to 0.2126 for the red value, 0.7152 for the green value and 0.0722 for the blue value, which gets results closer to color balance.

Командная строка Siril

asinh [-human] stretch [offset]

Преобразует изображение, используя функцию arcsin, благодаря чему становятся видны тусклые объекты. Обязательный аргумент команды — stretch, обычно в диапазоне от 1 до 1000, определяет интенсивность растягивания. Точка чёрного может быть смещена с помощью аргумента offset в нормализованном значении пикселя в диапазоне [0, 1]. Наконец, аргумент -human позволяет использовать весь на основе относительной чувствительности человеческого глаза для вычисления светимости, применяемой при расчёте растягивания для каждого пикселя, а не просто средние значения пикселей каналов. Этот метод растягивания сохраняет светимость в цветовом пространстве L*a*b*. 

8.1.2 Midtone Transfer Function Transformation (MTF)

MTF is one of the most powerful tools for stretching the image. It can be easily automated and that’s why the auto-stretched view uses it.

The tool is presented in the form of a histogram with 3 sliders (in the form of a triangle placed underneath) that we must move to transform the image. The triangle on the left represents the shadow signal, the one on the right the highlights and finally, the one in the middle the midtone balance parameter. The values of these sliders are displayed below the histogram, on the left, and can be changed directly by hand. Opposite is the percentage of pixels that are clipped by the transformation: it is important not to clip too many pixels. If only the midtones parameter is changed, then no pixel can be clipped.

The new pixel values are then computed with this function:

\[
MTF(x_p) = \frac{(m - 1)x_p}{(2m - 1)x_p - m} \quad (8.1)
\]

- For \(x_p = 0\), MTF = 0.
- For \(x_p = m\), MTF = 0.5.
- For \(x_p = 1\), MTF = 1.

where \(x_p\) is the pixel value defined as follow

\[
x_p = \frac{\text{original} - \text{shadows}}{\text{highlights} - \text{shadows}} \quad (8.2)
\]

Примечание: It is generally not recommended to change the value of the highlights, otherwise they will become saturated and information will be lost.
Рис. 3: Dialog box of the Histogram Transformation
The toolbar contains many buttons that affect the visualization of the histogram. You can choose to display the input histogram, the output histogram, the transfer curve and the grid. The button \[\text{autostretch}\] allows you to apply the same transformation as the autostretch algorithm. It is rarely advisable to use this button as is. Adjustments are usually necessary to avoid losing information. At the top of the histogram it is also possible to choose to display the histogram in logarithmic view, as in the illustration. This behavior can be made default as explained \textit{here}. Finally a zoom in X is available. This is very useful when all the signal is concentrated on the left of the histogram.

Командная строка Siril

```
mtf low mid high [channels]
```

Applies midtones transfer function to the current loaded image.

Three parameters are needed, \texttt{low}, \texttt{midtones} and \texttt{high} where midtones balance parameter defines a nonlinear histogram stretch in the [0,1] range.

Optionally the parameter \texttt{[channels]} may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

Примечание: \texttt{mtf} is also a function that can be used in the \textit{PixelMath} tool.

Командная строка Siril

```
autostretch [-linked] [shadowsclip [targetbg]]
```

Автоматически растягивает текущее загруженное изображение с различными параметрами для каждого канала (если каналы несвязаны), если только не передан аргумент \texttt{-linked}. Необязательные аргументы: \texttt{shadowclip} — точка обрезки теней, выраженная в значениях от основного пика гистограммы (по умолчанию -2.8); \texttt{targetbg} — целевое значение фона, в диапазоне [0, 1] (по умолчанию 0.25), определяющее окончательную яркость изображения.

\textbf{Applying transformation to the sequence}

This transformation can easily be applied to a sequence. You just have to define the transformation on the loaded image (with a sequence already loaded), then check the \textit{Apply to sequence} button and define the output prefix of the new sequence (\texttt{stretch}, by default), or use the following command:

Командная строка Siril

```
seqmtf sequencename low mid high [channels] [-prefix=]
```
Same command as MTF but for the sequence `sequencename`.

The output sequence name starts with the prefix "mtf_" unless otherwise specified with `-prefix=` option.

Optionally the parameter `[channels]` may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

Links: `mtf`

### 8.1.3 Generalised Hyperbolic Stretch transformations (GHS)

This is the most capable and modern tool of Siril, also the most complex to learn. A very detailed tutorial for this tool in Siril was written by the authors of this algorithm: [https://siril.org/tutorials/ghs](https://siril.org/tutorials/ghs). Here, we will just summarize here the basic operation of this tool.

Simply put, the GHS is able to improve the contrast of a range of brightness levels in an image. For example, if one wanted to better view the details in the medium to high brightness part of a nebula (which is in general very faint in an astronomy image), it would be possible to only select this range for stretching. It is very good at improving the contrast of main objects without making stars too big. The tool is very much based on iterative use, so stretching all the different ranges of brightness in the image one after the other, by small touches.

To achieve this, the tool relies heavily on histogram display and interaction, for each color channel. The transformation function, shaped like a hyperbole or an 'S', can be altered by moving its center (the SP - symmetry point parameter), by flattening either of its ends (with shadow and highlight protections), and of course its twist (stretch D and local stretch b factors). Manipulating these parameters on a small (for speed) image with an SP value of 0.5 will help you understanding their effect.

There are two main operations to do on each iteration: selecting the range of lights to modify, and actually modifying it. Selecting the range is quite easy, it's a matter of finding a representative value (SP) and defining the width of the range (b). Setting SP can be done in three ways:

- selecting an area of similar brightness in the image and clicking on the picker button
- clicking on the histogram itself with a single left click (it is possible to zoom in the histogram using the + button at the top left)
- using the cursor or its associated plus and minus buttons or direct value.

The width of the range depends on the local stretch. A high value of b will make a small range, and increase contrast over a small range of brightnesses in the image.

Modifying the histogram once the location of the change has been set is a more complex operation. One goal given by the algorithm's authors is to make the logarithmic view of the histogram (enabled by checking the box) as close as possible to a decreasing line. To do this, bumps need to be carved out and valleys to be filled. Here is a quick guide of values to use depending on what needs to be achieved:

- **initial stretch from linear**: set SP slightly to the left of the main peak, moderate b value from 6 and up, increase D slightly only to start to see the main object. Do not stretch too much at this point like an autostretch would do, otherwise the stars would grow too big ([main tutorial section for this](https://siril.org/tutorials/ghs)).

- **improving contrast of a range, or filling a valley**: set SP to the centre of the valley in the histogram, set b as high as how narrow the range or valley is, decrease HP to preserve stars, increase D slowly until the improvement appears.
Рис. 4: Dialog box of the Generalized Hyperbolic Stretch
• **decreasing contrast of a range, or flattening a peak**: decreasing a peak is not easy to do but will happen as a side effect of valleys being filled. For example, creating a peak, or filling a valley, will decrease what is on the left of SP. Another possibility is to use the inverse transformation, from the Type of stretch combo box, and a high LP value, and HP at 1.

• **move curve to the left, making the image darker**: often if we stretched the entire histogram, the peak will move to the right, making the background too bright. There is a simple way to just move everything to the left, select in the Type of stretch combo box the last entry, Linear stretch (BP shift). There's only one cursor to move now, controlling how much it will shift.

Some operations are also common for **color images**, where we often want to have a similar shape of curve for the three channels, working on each channel independently by unselecting them with the three colored circles below the histogram view:

![Generalized Hyperbolic Stretch Transformations](image)

Рис. 5: The Generalized Hyperbolic Stretch with a color image

• **moving the peak to the right**: a simple stretch with a SP value left of the peak will do that in general, so this should be done as part of a stretch.

• **spreading a peak**: to stretch a channel a bit more and it give it more importance in the final result, without changing the location of the peak too much, set SP near the peak or slightly to its right, set b depending on how the contribution is expected throughout the channel, between a negative value if the impact shall be felt up to the stars levels (to change their color) and a high value if this is only for a nebula, increase D to obtain the target width of the peak, and then offset the peak to the left by increasing HP.
• **moving all channels together:** an alternative luminance mapping stretch exists, see the *Color stretch model* combo box at the top right of the GHS window, using either luminance stretch values will stretch the luminance and reapply colors on it instead of stretching directly the three channels. The luminance modes can be better at preserving colours in the image. These modes use the same RGB blend clipping mode described above to prevent color channel clipping artefacts.

• **remapping image saturation:** the GHS transforms can be applied to the image saturation channel by selecting the Saturation option from the *Color stretch model* combo box. When this mode is selected the pre- and post- stretch saturation histograms will be shown in yellow. All the GHS options are available and this mode can provide highly targeted adjustment of the image saturation channel. A simple method of increasing the saturation in relatively unsaturated regions while preventing oversaturation is to use an **Inverse generalised hyperbolic transform** stretch with SP set to around 0.5, and HP brought down low enough to flatten the upper end of the saturation histogram.

Рис. 6: The image above shows how applying the GHS tool to the saturation channel gives an easy way of strongly enhancing saturation in a low-saturation image while still retaining control of the upper end of the saturation histogram, here used to create a 'Mineral Moon' image highlighting the differing mineral composition of different regions of the lunar surface.

Командная строка Siril

```
ght -D= [-B=] [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat] [channels]
```

Generalised hyperbolic stretch based on the work of the ghsastro.co.uk team.

The argument *-D=* defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

* B defines the intensity of the stretch near the focal point, between -5 and 15;
**LP** defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;

**SP** defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;

**HP** defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted B, LP and SP default to 0.0 ad HP defaults to 1.0.

An optional argument (either **-human**, **-even** or **-independent**) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument **-sat** specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work. Optionally the parameter [channels] may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels

**Командная строка Siril**

`invght -D= [-B=] [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat] [channels]`

Inverse generalised hyperbolic stretch based on the work of the ghsastro.co.uk team.

The argument **-D=** defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

**B** defines the intensity of the stretch near the focal point, between -5 and 15;

**LP** defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;

**SP** defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;

**HP** defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted B, LP and SP default to 0.0 ad HP defaults to 1.0.

An optional argument (either **-human**, **-even** or **-independent**) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument **-sat** specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work. Optionally the parameter [channels] may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels

**Командная строка Siril**

`modasinh -D= [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat] [channels]`

Modified arcsinh stretch based on the work of the ghsastro.co.uk team.
The argument -D= defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

LP defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;

SP defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;

HP defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted LP and SP default to 0.0 ad HP defaults to 1.0.

An optional argument (either -human, -even or -independent) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument -sat specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

Optionally the parameter [channels] may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

Командная строка Siril

```
invmodasinh -D= [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] [channels]
```

Inverse modified arcsinh stretch based on the work of the ghsastro.co.uk team.

The argument -D= defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

LP defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;

SP defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;

HP defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted LP and SP default to 0.0 ad HP defaults to 1.0.

An optional argument (either -human, -even or -independent) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument -sat specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

Optionally the parameter [channels] may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

Командная строка Siril

```
linstretch -BP= [-sat] [channels]
```

Stretches the image linearly to a new black point BP.
There is one mandatory argument:

**BP** provides the new black point to stretch to.

The argument [channels] may optionally be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

Optionally the parameter **-sat** may be used to apply the linear stretch to the image saturation channel. This argument only works if all channels are selected

## Applying transformation to the sequence

This transformation can easily be applied to a sequence. You just have to define the transformation on the loaded image (with a sequence already loaded), then check the *Apply to sequence* button and define the output prefix of the new sequence (*stretch_* by default). All of the commands have a sequence processing form too. Each sequence stretching command starts with *seq* and the first argument must be the sequence name, but they are otherwise the same.

Командная строка Siril

```plaintext
seqght sequence -D= [-B=} [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat] [-channels] [-prefix=]
```

Аналогична команде GHT, но имя последовательности должно быть указано в первом аргументе

Links: *ght*

Командная строка Siril

```plaintext
seqinvght sequence -D= [-B=} [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat] [-channels] [-prefix=]
```

Same command as INVGHT but the sequence must be specified as the first argument. In addition, the optional argument **-prefix=** can be used to set a custom prefix

Links: *invght*

Командная строка Siril

```plaintext
seqmodasinh sequence -D= [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat] [-channels] [-prefix=]
```

8.1. Image stretching
Same command as INVMODASINH but the sequence must be specified as the first argument. In addition, the optional argument `-prefix=` can be used to set a custom prefix

Links: invmodasinh

Komandnaya struktura Siril

```
seqinvmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] \n→ [channels] [-prefix=]
```

Same command as INVMODASINH but the sequence must be specified as the first argument. In addition, the optional argument `-prefix=` can be used to set a custom prefix

Links: invmodasinh

Komandnaya struktura Siril

```
seqlinstretch sequence -BP= [channels] [-sat] [-prefix=]
```

Same command as LINSTRETCH but the sequence must be specified as the first argument. In addition, the optional argument `-prefix=` can be used to set a custom prefix

Links: linstretch

8.2 Colors

8.2.1 Color Calibration

Siril offers two ways to retrieve the colors of your image. Here, "retrieve" means re-balancing the RGB channels to get as close as possible to the true colors of the shot object.
**Manual Color Calibration**

| Предупреждение: | The color calibration must be performed on a linear image whose histogram has not yet been stretched. Otherwise, the colors obtained are not guaranteed to be correct. |

The manual way uses the following window:

The first step deals with the **background** of your image. The goal is to equalize the RGB layers in order the background appears as a neutral grey color.

After making a selection in your image (in a not so crowdy nor contrasted area), the area is taken into account by clicking on the *Use current selection* button. The coordinates of the rectangle are displayed. Then *Background Neutralization* will calculate the median of each channel and equalize them.

The second step deals with the **bright objects** of the picture. You can modify once again the histogram in two ways:

- Manually, with *White reference* and the 3 R, G and B coefficients, according to your own taste.
- Automatically, by selecting a rectangle area with contrasted objects (the same way as previously)

Two sliders allow you to change the rejection limit for too dark and too bright pixels in the selection.

As this is a trial and error process, you can undo the result with the *Undo* button (up left) and then try with other selections or coefficients until you are satisfied.

**Photometric Color Calibration**

| Предупреждение: | The calibration of the colors by photometry must imperatively be carried out on a linear image whose histogram was not yet stretched. Without what, the photometric measurements will be wrong and the colors obtained without guarantee of being correct. |

Another way for retrieving the colors is to compare the color of the stars in the image with their color in catalogues, to obtain the most natural color in an automatic and non-subjective way. This is the PCC (Photometric Color Calibration) tool. It can only work for images taken with a set of red, green and blue filters for colors, or on-sensor color. To identify stars within the image with those of the catalogue, an astrometric solution is required. Running the PCC tool will first do that, so for this first part, please see the documentation of the plate solver module.

| Примечание: | This technique is heavily dependent on the type of filter used. Using different kinds of R, G, B filters will not make a large difference, but using a Light pollution filter or no IR-block filters will make the solution deviate significantly and not give the expected colors. |

Since version 1.2, the two tools run independently: it is possible to run the photometric analysis and color correction of the image only if the image has been already plate solved. It also means different catalogues can be used for PCC and astrometry. And now, the tool is also available as the pcc command, so it can be embedded in image post-processing scripts.

If the image was previously plate solved, turn on the *annotations* feature to check that catalogues align with the image. If the astrometric solution is not good enough, checking *Force plate solving* will force its recomputation as part of the PCC process.

As a reminder from the plate solver documentation, here is a summary of the options visible in the window:
Рис. 7: Manual Color Calibration dialog window.
Рис. 8: Photometric Color Calibration dialog window.
• Make sure the sampling is correct, computed from the focal length and pixel size found in the image or copied from the settings.

• The *Flip image if needed* allows to reorient the image correctly according to the astrometry result.

• For some oversampled or too large images, it is useful to check the *Downsample image* to have more chances of success with the plate solving and it's also faster.

• The *Auto-crop (for wide field)* option will limit the field to 5 degrees in case you are dealing with very wide field images, mostly useful for plate solving.

• The *Catalog Settings* section allows you to choose which photometric catalog should be used, NOMAD or APASS, as well as the limiting magnitude.

**Совет:** The NOMAD catalog can be *installed locally*, while the APASS catalogue needs an internet access to get its content.

• The *Star Detection* section allows you to manually select which stars will be used for the photometry analysis. It's better to have hundreds of them at least, so individual picking would not be ideal.

• If desired, the *Background Reference* can be manually selected as described in *Manual Color Calibration*. This can be useful in the case of nebula images where the background sky parts are small.

When enough stars are found and the astrometric solution is correct, the PCC will print this kind of text in the Console tab:

```
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
```

We can understand that 433 stars were selected from the catalogue and the image for photometric analysis, but somehow, only 363 we actually used, 70 being excluded. The line *Distribution of errors* explains for what reason they were excluded: 18 were not found in the expected position, 48 were too big and 4 probably
saturated. It is very common to have many stars rejected because they don’t meet the strict requirements for a valid photometric analysis.

We can also see that the PCC found three coefficients to apply to the color channels to correct the white balance. The deviation here, which is the average absolute deviation of the color correction for each of the star of the photometric set, is moderately high. On well calibrated images without gradient, with correct filters and without a color nebula covering the whole image, deviation would get closer to 0.

Командная строка Siril


Run the Photometric Color Correction on the loaded image.

If the image has already been plate solved, the PCC can reuse the astrometric solution, otherwise, or if WCS or other image metadata is erroneous or missing, arguments for the plate solving must be passed: the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space.

focal length and pixel size can be passed with `-focal=` (in mm) and `-pix\_size=` (in microns), overriding values from image and settings.

you can force the plate solving to be remade using the `-platesolve` flag.

Unless `-noflip` is specified and the image is detected as being upside-down, the image will be flipped if a plate solving is run.

For faster star detection in big images, downsampling the image is possible with `-downscale`.

The limit magnitude of stars used for plate solving and PCC is automatically computed from the size of the field of view, but can be altered by passing a +offset or -offset value to `-limitmag=`, or simply an absolute positive value for the limit magnitude.

The star catalog used is NOMAD by default, it can be changed by providing `-catalog=apass`. If installed locally, the remote NOMAD can be forced by providing `-catalog=nomad`

Links: nomad

### 8.2.2 Color Saturation

This tool is used to increase the color saturation of the image. It is possible to choose between a specific hue or the global hue to enhance. The strength of the saturation is adjusted with the slider `Amount`.

The `Background factor` slider sets the factor multiplied by the background value. Lower is the value, stronger is the saturation effect. While a high value will preserve the background.

Командная строка Siril

\[satu\ amount\ [\text{background\_factor}\ [\text{hue\_range\_index}]]\]
8.2.3 Remove Green Noise

Because green is not naturally present in deep sky images (except for comets and some planetary nebulae), if the image has already been calibrated, its colors are well balanced and the image is free of any gradient, we can assume that if the image contains green, it belongs to the noise. It is then interesting to find a method to remove this dominant green. This is exactly what the Remove Green Noise tool proposes, which is derived from the Subtractive Color Noise Reduction tool, but for green only.

This tool has 3 settings. The protection method, the amount (called $a$ in the following section), and a Preserve lightness button. The following methods present the different existing ways to remove the green pixels by replacing them with a mix of Red and Blue. The amount is only available for methods with mask protection. The choice of its value must be done with caution in order to minimize the rise of the magenta cast in the sky background. Do not hesitate to use the Undo and Redo buttons in order to fine-tune the value.
Protection method

Maximum Mask Protection

\[ m = \max(R, B) \]
\[ G' = G \times (1-a) \times (1-m) + m \times G \]

Additive Mask Protection

\[ m = \min(1, R + B) \]
\[ G' = G \times (1-a) \times (1-m) + m \times G \]

Average Neutral Protection (default method)

\[ m = 0.5 \times (R + B) \]
\[ G' = \min(G, m) \]

Maximum Neutral Protection

\[ m = \max(R, B) \]
\[ G' = \min(G, m) \]

Finally, the \textit{Preserve lightness} button preserves the original CIE L* component in the processed image, in order to only process chromatic component, it is highly recommended to let this option checked.

Командная строка Siril

\texttt{rmgreen [-nopreserve] [type] [amount]}
Applies a chromatic noise reduction filter. It removes green tint in the current image. This filter is based on PixInsight's SCNR and it is also the same filter used by HLVG plugin in Photoshop. Lightness is preserved by default but this can be disabled with the \texttt{-nopreserve} switch.

**Type** can take values 0 for average neutral, 1 for maximum neutral, 2 for maximum mask, 3 for additive mask, defaulting to 0. The last two can take an \texttt{amount} argument, a value between 0 and 1, defaulting to 1.

\section*{8.2.4 Negative Transform}

Negative transformation refers to subtracting pixel values from $(L_1)$, where $L$ is the maximum possible value of the pixel, and replacing it with the result.

The \textit{Negative transformation} tool is different from the negative view \includegraphics[width=1cm]{negative.png} in the toolbar. Indeed, the transformation is not only visual, but actually applied to the pixel values. If you save the image, it will be saved as a negative.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{original_image_with_weak_signal.png}
\caption{Original image with weak signal (Image Cyril Richard).}
\end{figure}

\textbf{Совет:} A common use of the negative transformation tool is to remove the magenta cast from SHO images. In this case one need to apply \textit{Negative transformation}, then \textit{Remove Green Noise}, then \textit{Negative transformation}.\hspace{0.5cm}
Рис. 12: Negative image where the signal is more visible (Image Cyril Richard).
8.3 Фильтры

В этом разделе описаны фильтры, доступные в Siril. Фильтры — это инструменты, которые изменяют пиксели изображения в соответствии с потребностями.

8.3.1 À Trous Wavelets Transform

A wavelet is a function at the base of the wavelet decomposition, a decomposition similar to the short term Fourier transform, used in signal processing. It corresponds to the intuitive idea of a function corresponding to a small oscillation, hence its name.

There are many types of wavelet functions that have their own names, as shown in the figure below.

The À Trou Wavelet Transform used in Siril performs decomposition of an image into a series of scale layers, also known as wavelet layers. These layers can be extracted with the Wavelet Layers extraction tool, however here, they are used without being visually accessible. In general, this algorithm is widely used at the end of a planetary image stack. Because the noise is exclusively contained in one of the wavelet layers, it is possible to bring out the details of the image by containing the noise amount.

The first thing to do is to click on the Execute button in order to calculate the wavelet layers using the parameters defined above, such as:

- **Type**: There are two types of algorithms possible: Linear and BSpline. The latter will usually be chosen, even if it is a bit slower.

- **Nb of layers**: Number of wavelet layers that will be used. 6 is the maximum number of layers that can be defined. To work on a larger number of layers it is possible to use the command line explained below.

Then, each layer has a slider that allows to modify the contrast of this layer. If less than 6 layers have been created, then only the corresponding sliders will be active. A value greater than 1 improves the details while a smaller value tends to reduce them.

This is a liveview tool. The changes are displayed in real time and you have to click on Apply to validate them. Clicking on Reset resets all the sliders to 1, and thus cancels any transformation in progress.
Рис. 13: An example of four different types of wavelets.
Рис. 14: Wavelet tool dialog box.
Рис. 15: Wavelets applied on a Jupiter image (courtesy of J.-L. Dauvergne). The image on the left is the raw image of the stacking output, while the image on the right is the same image on which wavelets are applied.

```
wavelet nbr_layers type
```

Computes the wavelet transform on (nbr_layers=1...6) layer(s) using linear (type=1) or bspline (type=2) version of the 'a trous' algorithm. The result is stored in a file as a structure containing the layers, ready for weighted reconstruction with WRECONS

Links: wrecons

Командная строка Siril

```
wrecons c1 c2 c3 ...
```

Reconstructs to current image from the layers previously computed with wavelets and weighted with coefficients \(c_1, c_2, \ldots, c_n\) according to the number of layers used for wavelet transform

The example given in the image above would be written in the command line as follow:

```
wavelet 6 2
wrecons 31 5 1 1 1 1
```
8.3.2 Уменьшение полосения

In some cases, images may suffer from a banding defect. This is usually caused by the sensor and calibration by darks, bias and flats do not improve the images.

![Original image with visible banding.](image)

The banding reduction window dialog has some parameters to optimize the processing:

- **Amount** defines the strength of the correction. The higher the value, the stronger the correction.
- **Protect from Highlights** will ignore bright pixels when the option is checked.
- **1/Sigma Factor** will adjust the highlight protection. Higher value will give a better protection.
- **Vertical banding** allows user to fix banding if bands are vertical.

Applying the following filter to the original image, with parameter values as shown in the illustration, gives a nice result free of banding.

This transformation can easily be applied to a sequence. You just have to define the transformation on 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).

Командная строка Siril

```bash
fixbanding amount sigma [-vertical]
```
Рис. 17: Banding Reduction dialog box.

Рис. 18: Result after the filter has been run. No more bandings are visible.
Tries to remove the canon banding.

Argument **amount** defines the amount of correction.  
**Sigma** defines a protection level of the algorithm, higher sigma gives higher protection.  
**-vertical** option enables to perform vertical banding removal.

Командная строка Siril

```
seqfixbanding sequencename amount sigma [-prefix=} [-vertical]
```

Same command as FIXBANDING but for the sequence **sequencename**.

The output sequence name starts with the prefix "unband_ " unless otherwise specified with **-prefix=** option.  
**-vertical** option enables to perform vertical banding removal

Links: **fixbanding**

### 8.3.3 Contrast-Limited Adaptive Histogram Equalization (CLAHE)

The CLAHE method is used to improve the contrast of images. It differs from ordinary histogram equalization in that the adaptive method calculates multiple histograms, each corresponding to a separate section of the image, and uses them to redistribute the brightness values of the image. It can therefore improve local contrast and enhance edge definition in each region of an image.

**Совет:** This filter is a liveview filter. In other words, every change in the settings is automatically visible on the screen, but this can be disabled by unchecking the **Preview** button.

- The size of the tiles, in which the histograms are calculated, can be defined via a slider. By default it is set to 8.
- The Clip Limit is the option that prevents to overamplify noise in relatively homogeneous regions of an image. Then, the clipped part of the histogram that exceeds the clipping limit is redistributed equally among all the bins of the histogram.

**Совет:** This filter works better on non-linear data. It is recommended to stretch the image before.

Командная строка Siril

```
clahe cliplimit tileSize
```

Рис. 20: An example of CLAHE filter applied to a non-linear data with Tiles Grid Size=21 and Clip Limit=4.20.
Equalizes the histogram of an image using Contrast Limited Adaptive Histogram Equalization.

cliplimit sets the threshold for contrast limiting.

tilesize sets the size of grid for histogram equalization. Input image will be divided into equally sized rectangular tiles

8.3.4 Косметическая коррекция

In Siril, the cosmetic correction is the step that gets rid of hot and cold pixels in the image. It is usually done during pre-processing using the master-dark. This is because the latter usually contains a good map of the defective pixels and it is easier to find them on it. However, when you don't have a master-dark, Siril offers an alternative with an automatic detection algorithm of these pixels in a light image.

Рис. 21: Dialog window of Correction Cosmetic tool.

The dialog window contains several parameters necessary for the proper functioning of the tool. However, using the default settings usually gives good results.

- **Cold Sigma**: How many times (in average deviation units) a pixel value must differ from the value of surrounding neighbors to be considered as a cold pixel.

- **Hot Sigma**: How many times (in average deviation units) a pixel value must differ from the value of surrounding neighbors to be considered as a hot pixel.

- **Amount**: This is a modulation parameters where 0 means no correction and 1, 100% corrected.

- **CFA**: This option needs to be checked for CFA images with Bayer pattern. It does not work for X-Trans sensor.

This operation can be applied to sequences. Open a sequence and prepare the settings you want to use, then check the **Apply to sequence** button and define the output prefix of the new sequence (cc_ by default).

**Hot pixels detection**

Let's call $m_{5\times5}$ the median of the 5 nearest neighbors. If the pixel value is greater than

$$m_{5\times5} + \max(\text{avgDev}, \sigma_{\text{high}} \times \text{avgDev}),$$

with avgDev, the **Average Deviation** of the whole image.

Then the pixel is replaced by the average of the $3 \times 3$ pixels: $a_{3\times3}$, but only if

$$a_{3\times3} < m_{5\times5} + \frac{\text{avgDev}}{2}.$$
Cold pixels detection

If the pixel value is less than

\[ m_{5 \times 5} - (\sigma_{\text{low}} \times \text{avgDev}) \]

then the pixel is replaced by \( m_{5 \times 5} \).

Рис. 22: Animation showing cosmetic correction.

Командная строка Siril

```
find_cosme cold_sigma hot_sigma
```

Применяет автоматическое определение холодных и горячих пикселей с использованием порогов, указанных в аргументах

Командная строка Siril

```
find_cosme_cfa cold_sigma hot_sigma
```

Аналогична команде FIND_COSME, но для монохромных изображений CFA

Links: `find_cosme`

### 8.3.5 Реконструкция

Реконструкция это математический инструмент для компенсации эффектов размытия или искажения на изображении. Истинная сцена это не то что записано вашим сенсором — ваша запись это оценка истинной сцены, свёрнутой с помощью PSF (в математическом выражении "размывающая PSF" представляет искажения атмосферы, физические свойства вашего телескопа, съёмка при движении и т.д., которые ухудшают вашу оценку). С помощью реконструкции можно, в некоторой степени, обратить подобную деградацию изображения. Однако важно сразу отметить, что реконструкция это то, что математики называют некорректно поставленной задачей (как и большинство обратных задач). Это значит что решения может не существовать. Если решение существует, оно может не быть уникальным и может не иметь непрерывной зависимости от данных. По сути, это означает, что реконструкция, даже теоретически, действительно сложна, и нет никаких гарантий, что она сработает.

Всё становится ещё труднее когда мы точно не знаем, что представляет собой PSF, которую мы пытаемся удалить. В астрономии мы можем, в теории, получить представление о PSF по эффекту размытия точечных источников (звёзд), которые мы снимаем. Однако иногда истинная PSF непостоянная по всему изображению, иногда иные факторы, как насыщенность звёзд не позволяют PSF, полученной от звёзд, быть полностью точной оценкой PSF, а иногда, например при съёмке Луны, на изображении нет звёзд.

8.3. Фильтры
Siril стремится обеспечить гибкий подход к реконструкции. Имеются несколько настроек для определения или оценки PSF и несколько алгоритмов реконструкции для заключительного этапа реконструкции после того, как PSF определена.

Рис. 23: Пример реконструкции звёздного поля.

Реконструкция доступна из меню Обработка или с помощью команд Siril.

Обзор использования

- To generate a deconvolution PSF, select the required PSF generation method and press Generate PSF. This can be performed separately from the actual deconvolution so that the user can see the effect of changing the PSF parameters.

- Siril создаёт только монохромные PSF, поскольку это самый частый способ использования и упрощает пользовательский интерфейс. Однако можно сохранить и объединить 3 монохромных PSF, чтобы создать трёхканальную PSF, которая может быть загружена и использована для реконструкции трёхканальных изображений.

- To apply deconvolution to a single image, select the required PSF generation method and press Apply. If a blind PSF estimation method has previously been run, the method will automatically be set to Previous PSF, in order to avoid unnecessarily recalculating the PSF.

- Чтобы применить реконструкцию к последовательности, действуйте как описано выше, но убедитесь, что выбран пункт Применить к последовательности. Вы можете так же указать пользовательский префикс новой последовательности. Если не указано иное, по умолчанию будет использован префикс dec_.

- При реконструировании последовательности, PSF будет рассчитана только для первого изображения в последовательности. Одинаковая PSF будет повторно использована для всех изображений в последовательности.

Overview of Blur Kernel Definition Methods

- 0 Descent: This is the default PSF estimation method based on work by Anger, Delbracio and Facciolo. The parameters do not generally require adjustment, except that for particularly large PSFs you may wish to try the multiscale PSF estimation model. Multiscale is off by default as during development it was noted to have a tendency to produce rather unnatural results with the more common small to medium PSF sizes.

- Spectral Irregularities [Goldstein2012]: This PSF estimation method is offered as an alternative. In general it does not perform as well as the 0 descent method, however it may be useful if you discover an image where the default method does not give good results. For this method the latent sharp image needs not to contain any edges as long as the spectral decay model is respected. On the other hand, the 0 descent assumes a similar model (since edges have the same spectral decay), but requires to have sparse gradients and be contrasted, thus edges to be in-phase, so theoretically this model may work better on low contrast starless images. Some experimentation is likely required to find the algorithm that best fits your data.

- PSF From Stars: This method models a PSF from the average PSF of the selected stars. It is important to be selective about the stars you choose: they must not be saturated as that would give a gross distortion of the PSF estimate, but they must also not be so faint that Siril’s star analysis functions provide inaccurate measurements of the stars. The stars chosen should be reasonably bright, fairly central to the image and in an area of the image with a fairly constant background. Once stars are selected, you can pick either a Gaussian or Moffat star profile model and when executing the
Рис. 24: Диалоговое окно инструмента реконструкции.
deconvolution the PSF will be synthesized from the average parameters of the selected stars. If no stars are selected, Siril will attempt to autodetect stars with a peak amplitude between 0.07 and 0.7, with a Moffat profile. This range avoids saturated stars as well as those that are too faint to give an accurate solution, and generally provides good results.

- **Manual PSF**: This method allows you to define a PSF manually. Gaussian, Moffat or disc PSF models can be defined. Note that the FWHM is specified in pixels, not arc seconds. The Gaussian and Moffat models are suitable for deconvolving the shapes of stars resulting from atmospheric distortion; the disc PSF model is suitable for deconvolving the effect of being slightly out of focus.

- **Load PSF from file**: This method allows you to load a PSF from any image format supported by Siril. The provided PSF must be square (it will be rejected if not square) and should be odd (it will be cropped by one pixel in each direction if not odd, but this will give a slightly off-centre PSF and is not optimal compared with providing an odd PSF in the first place). Either monochrome or 3-channel PSFs may be loaded. If a 3-channel PSF is loaded in conjunction with a monochrome image, the evenly-weighted luminance values of the PSF will be used. If a 3-channel PSF is loaded together with a 3-channel image then each channel of the image will be deconvolved using the corresponding channel of the PSF. If a monochrome PSF is loaded together with a 3-channel image then the image will be converted to the LAB colour space and the L (Luminance) channel will be deconvolved using the monochrome PSF for computational efficiency, and the deconvolved L will be recombined with the A and B channels and converted back to RGB.

- **Previous PSF**: This method allows reuse of the previously estimated PSF. It is mostly of use with the blind PSF estimation methods: if you are content with the estimated PSF but wish to make a number of test runs using different parameters for the final stage of deconvolution, you can reuse the previous PSF and save some computation time.

- **Once estimated, PSFs may be saved if desired. If Siril is compiled with libtiff support then the PSF will be saved in 32-bit TIFF format, with the same filename as the current image but date-and-time-stamped and suffixed with _PSF. If Siril has been built without libtiff support, the PSF will be saved as a FITS file. While this is Siril’s primary format for astronomical image files, TIFF is preferred for PSFs: the disadvantage of using the FITS format for PSFs is potential reduced compatibility with image editors that you may wish to use to edit or examine the saved file.

**Совет**: If the blind generation of a deconvolution PSF can be done on linear and non-linear data, the use of a PSF from star PSF can only be done on linear images. Otherwise the PSF values would not be valid.

**Overview of Non-Blind Deconvolution**

- **Richardson-Lucy Deconvolution** [Lucy1974]: This is the default non-blind deconvolution algorithm. It is an iterative method, famous for its use in correcting image distortions in the early operating period of the Hubble Space Telescope, and in Siril is regularized using either the Total Variation method, which aims to penalize the algorithm for amplifying noise, or the Frobenius norm of the local Hessian matrix. This regularization is based on second derivatives. As well as regularization an early stopping parameter is provided, which allows the algorithm to be halted early once its rate of convergence falls below a certain level. Increasing the value of the early stopping parameter can reduce ringing around stars and sharp edges. Two formulations of the Richardson-Lucy algorithm are provided: the multiplicative formulation and the gradient descent formulation. The latter can allow better control, as the gradient descent step size can be altered (the downside of this is that by using more small steps, more iterations are required to reach the same level of convergence). The bigger advantage of the gradient descent method is that it allows more regularization to be used - this can be problematic in the multiplicative Richardson-Lucy algorithm as the regularization term appears in the denominator and small values here (strong regularization) can cause instability. Siril will use naive convolution for small kernel sizes
and FFT-based convolution for larger kernel sizes where FFTs provide a more efficient algorithm. (This is automatic and requires no user intervention.)

- **Wiener Filtering Method**: This method is a non-iterative deconvolution method. It models an assumed Gaussian noise profile, i.e. noise modelled by a constant profile. The constant alpha is used to set the regularization strength in relation to the noise level. As with the other algorithms, a smaller value of alpha provides more regularization. This algorithm can be good for lunar images where the noise regime is Gaussian not Poisson, but usually works badly on deep space imagery where the noise still tends to have a Poisson character.

- **Split Bregman Method**: This method is used internally within the blur PSF estimation processes, and is also offered as a final stage deconvolution algorithm. It is a commonly used algorithm in solving convex optimization problems. This algorithm is also regularized using a total variation cost function. It does not perform as well as Richardson-Lucy on starscapes but may be considered for starless images or lunar surface images.

**Совет**: Choice of deconvolution method is very important to obtaining good results. Generally for DSO images it is important to use a Richardson-Lucy method: both the Split Bregman and Wiener methods give poor results around stars because of the extreme dynamic range. For linear images it is usually best to use the gradient descent Richardson-Lucy methods, and if ringing occurs around bright stars then reduce the step size. This approach reduces the impact of each iteration therefore more iterations are required, but it does mean that you can achieve finer control taking deconvolution just up to the point where artefacts start to form and then backing off very slightly. For stretched images the multiplicative Richardson-Lucy algorithms may be used.

**Совет**: For stacked lunar and planetary images the Split Bregman or Wiener methods can be more appropriate. These methods do not generally require iteration in the way that Richardson-Lucy does, and they may be better suited to the noise characteristics of stacked, high signal-to-noise ratio images. (The Richardson-Lucy algorithm is based on an assumption of Poisson noise, which is usually true for DSO imaging, whereas the Wiener method implemented here assumes a Gaussian noise distribution which may fit stacked planetary / lunar images better).

**Parameters and Settings**

**Общие настройки**

- **PSF size**: The input PSF size should be chosen sufficiently large to assure that the PSF is included in the specified domain. However, setting it too large can result in a poorer and more time-consuming result from the blind PSF estimation methods.

- **Lambda (\(\lambda\))**: Regularization parameter for PSF estimation. Try decreasing this value for noisy images.
**Descent PSF estimation settings**

- **Multiscale.** This setting enables multiscale PSF estimation. This may help to stabilize the PSF estimate when specifying a large PSF size, but some PSFs generated with this option can give rise to unnatural looking results and it is therefore off by default.

- **Expert settings.** These should not normally require adjustment, but are made available for the curious.
  - **Gamma** sets the regularization strength used when carrying out the sharp image prediction step. For a given gamma, as the noise increases the estimation also gets noisier. If gamma is increased, the estimation is less affected by noise but tends to be smoother. The default value of 20 was determined experimentally in [Anger2019].
  - **Iterations** sets the number of iterations used in the PSF estimate procedure. The authors of the algorithm report that there is minimal benefit in increasing this to 3 and no benefit at all in increasing it beyond 3.
  - **Lambda ratio and lambda minimum** sets the parameters for refining the sharp image prediction through successive values of the sharp image predictor regularization parameter at each iteration of the method.
  - **Scale factor, upscale blur and downscale blur** are only used when multiscale estimation is active. These set the default scale factor between each scale level and the amount of blurring to use when rescaling between each scale.
  - **Kernel threshold.** Values below this level in the PSF estimate are set to zero.

**Spectral Irregularity PSF estimation settings**

- **Compensation factor** controls the strength of a filter used to avoid excessive sharpness in the estimated PSF. For images with intrinsic blur, a value close to unity should be used. For intrinsically sharp images, low values can result in artefacts and the value should be increased to a large number, effectively disabling the filter.

- **Expert settings.** These should not normally require adjustment, but are made available for the curious.
  - **Inner loop iterations** sets the number of iterations performed in the inner loop of the spectral irregularity method. The algorithm converges quickly and it may be possible to reduce this to approximately 100 without much degradation of the result.
  - **Samples per outer loop.** This controls how many random phases should be sampled. Because the phase retrieval starts with random values for each sample, it is important to draw enough samples to avoid converging to a local minimum. The PSF stabilizes quickly for low noise images, but if looking for improved results from this method, this is the first of the expert settings to try adjusting especially with images with higher noise levels.
  - **Outer loop iterations.** [Anger2018], suggests that 2 iterations can be enough to produce a plausible PSF estimate, and there is negligible value in increasing this above 3.
PSF по звёздам

- This PSF generation method has no adjustable parameters. It generates a PSF based on the average parameters of the selected stars, using the findstar command or the Dynamic PSF dialog. The average parameters are shown in the deconvolution dialog when this PSF generation method is selected. It is preferable for the user to actively select the stars they wish to use for this method, to obtain the most accurate PSF. Ideally around 10 fairly bright but not saturated stars should be selected from the central region of the image (to exclude stars that may suffer from coma or other aberrations). However, if the user has not selected any stars, Siril will attempt to autodetect suitable stars by running its detection routine with filters set to keep only stars with peak amplitudes between 0.07 and 0.7. This range avoids both saturated stars and those that are too faint to give an accurate solution. It will work well in most cases but may still be affected by off-centre aberrations.

- If you select the Symmetrical PSF checkbox, the generated PSF will be circular. This will match the average FWHM and beta of the selected stars but will not match any elongation.

PSF вручную

This PSF generation method allows specification of a custom parametric PSF.

- Profile type allows choice of PSF profile. Gaussian, Moffat, disc and Airy disc PSFs are supported.
  - Gaussian and Moffat PSFs are used for matching star parameters measured from the image. They should provide a good estimate of the total blur function being applied to the image, as stars are point sources.

  Рис. 25: An example of Moffat PSF with fwhm=5", Angle=45°, Ratio=1.20, β = 4.5 and a PSF size of 15.
  - Disk PSFs are used to deconvolve images that are out of focus.

  Рис. 26: An example of Disk profile with fwhm=5" and a PSF size of 15.
  - Airy disc PSFs are used to deconvolve the diffraction that arises as a physical consequence of diffraction by the aperture of your telescope.

  Рис. 27: An example of an Airy-Disk PSF with Diameter=250mm, Focal Length=4500mm, Wavelength=525nm, Pixel Size=2.9μm, Central Obstruction=40% and a PSF size of 41.

- FWHM specifies the full width at half maximum of the chosen profile (for disc PSFs it simply specifies the radius).
- Beta (β) specifies the beta parameter used in the Moffat PSF profile. It is ignored for other PSF profiles.
- For Airy disc PSFs a number of parameters of your telescope and sensor are required:
– Aperture
– Focal length
– Sensor pixel size
– Central wavelength being imaged. Siril will try to extract this data from your image metadata where available, but if some parameters are missing or look unreasonable Siril will highlight them and print a warning in the log recommending you check them. The ratio of the central obstruction is also required to generate an accurate Airy disc. This is expressed as a percentage, i.e. the total area of the central obstruction divided by the total area of the aperture x 100. For refractors this is zero; for other telescopes it varies: it may be around 20% for a Newtonian reflector or as much as 40-50% for some Corrected Dall-Kirkham telescopes. You will need to measure your instrument or consult the manufacturer’s specifications.

**Richardson-Lucy Deconvolution**

The parameters used to configure Richardson-Lucy deconvolution in Siril are as follows:

- **alpha** sets the regularization strength. A smaller value of alpha gives stronger regularization and a smoother result; a larger value reduces the regularization strength and preserves more image detail, but may result in the amplification of noise.

- **Iterations** specifies the maximum number of iterations to use. In the absence of noise, a large number of iterations will cause deconvolution to converge the estimate closer to the true image, however an excessively large number of iterations will also magnify noise and cause ringing artefacts around stars. The default is 1 iteration: a higher number can be set to compute multiple iterations automatically, or you can keep pressing **Apply** to apply one iteration at a time until you are happy with the result. (Or go one further, decide you’re no longer happy and use **Undo**.)

- **Stopping criterion** sets a convergence criterion based on successive estimate differences. This will stop the algorithm once convergence is within the specified limit. This is an important parameter - if you are getting rings around stars in your final image, try increasing the value of the stopping criterion. This may be disabled altogether using the check button.

- **Algorithm method** specifies whether to use the multiplicative implementation or the gradient descent implementation.

- **Step size** specifies the step size to use for the gradient descent implementation. Do not set it too large or the algorithm will not converge. This parameter has no effect if the multiplicative implementation is selected.

_Cooper:_ For linear images, try using the gradient descent methods provide the control necessary to prevent ringing around stars. For deconvolving stretched images, however, this can be unnecessarily slow, so using the multiplicative methods can often save time without compromising image quality.
Split Bregman Deconvolution

The parameters used to configure Split Bregman deconvolution in Siril are as follows:

- **alpha** sets the regularization strength. A smaller value of alpha gives stronger regularization and a smoother result; a larger value reduces the regularization strength and preserves more image detail, but may result in the amplification of noise.

- **Iterations** specifies the maximum number of iterations to use. The Split Bregman method does not require multiple iterations in the form implemented here, but may be iterated if desired. This generally makes only a small difference and therefore defaults to 1.

Wiener Deconvolution

Wiener deconvolution in Siril only requires one parameter:

- **alpha** sets the regularization strength. A smaller value of alpha gives stronger regularization and a smoother result; a larger value reduces the regularization strength and preserves more image detail, but may result in the amplification of noise.

FFTW Performance Settings

The PSF estimation and deconvolution algorithms make extensive use of fast Fourier transforms using the FFTW library. This offers a number of tuning options, which can be adjusted in the performance tab of the main Siril preferences dialog.

Note on Image Row Order

Different types of image processed by Siril can have their pixel data arranged in different orders. SER video files always store data top down, whereas FITS files may store data either bottom up or top down. Bottom up is the original recommendation, however increasingly FITS are sourced from CMOS cameras which tend to follow a top down pixel order.

When an image is deconvolved with a PSF created from the same image (or with it open) this causes no problem. However there is potential for problems to arise if a PSF is generated with an image with one row order and used to deconvolve an image or sequence with the opposite row order. This is a niche use case, but handling it consistently results in behaviour which at first sight can be surprising; it is therefore explained below.

Siril handles the issue by tracking the row order of the image the PSF was created with. PSFs are always saved using a bottom up row order (automatically flipping them if they were created with a top down image), and when loaded the row order is matched to the row order of the currently open image. If an image of the opposite row order is opened, the row order of the PSF will be changed to match. This means that if, for example, you take some bottom up FITS images, use one of them to generate a PSF, and then convert them to a top down SER sequence, the PSF will be converted to the correct orientation to match the SER sequence. If a PSF is being previewed at the time an image with the opposite row order is opened the preview will not update immediately; the row order change will be detected automatically and the PSF flipped at the time when it is applied to the image.
**Rogues' Gallery**

This section shows some examples of where deconvolution has gone wrong, together with explanations of why.

![Image of deconvolution tool]

Рис. 28: The manually specified PSF was too big, resulting in large dark rings around stars.

**Deconvolution: Usage Tips**

You've arrived here from the hints button in the deconvolution tool in Siril. No worries: deconvolution is a tricky technique. Even theoretically, it's really hard: there are no guarantees the maths will always converge to a unique solution that improves your image. That said, here are some tips to help you get the most out of Siril's deconvolution algorithms.

**What PSF to use?**

Using an accurate PSF is fundamental to achieving good results from deconvolution. The two simplest ways to generate a PSF are to use a blind PSF estimate, or to model your PSF on stars in your image.

**PSF по звёздам**

Siril can detect and model stars in your image. See the Dynamic PSF help page for details. To get a good model for your PSF, try selecting the Moffat star profile in Dynamic PSF. Stars are point sources so the spread function of an average star is a good model for the blurring effects that we are trying to remove by deconvolution.

**Совет:** Once you have detected stars, sort them by peak amplitude (parameter “A”). Select and delete any with amplitude greater than 0.7 or less than 0.1, and if your image contains background galaxies check
Рис. 29: Too many iterations have been applied. (I applied them one at a time to exaggerate the result, which is why the iterations parameter still says 1.)

that no false positives remain. Stars in this brightness range are not saturated and not too faint to give an accurate PSF model.

**Совет:** If the blind generation of a deconvolution PSF can be done on linear and non-linear data, the use of a PSF from star PSF can only be done on linear images. Otherwise the PSF values would not be valid.

**Blind PSF Estimate**

These methods can automatically estimate a PSF based on the image itself. If you have no better prior knowledge of the PSF such as stars in the image (for example, lunar imagery that contains no stars) then this may be your best option. In most cases it is recommended to use the default method: it is faster and usually gives better results.

**Совет:** However you are generating your PSF, check the preview to make sure that it does not look cropped. If it does, increase the PSF size until no significant parts of the PSF are cropped.
Рис. 30: Close-up showing the effect of trying to apply too much regularization (alpha = 30) using the multiplicative version of Richardson-Lucy. For strong regularization and / or better control over each iteration, the gradient descent formulation is recommended.
Рис. 31: Typical example of attempting to deconvolve an unstretched starfield using Split Bregman (in this case) or Wiener filter deconvolution. Those are better for planetary / lunar / solar images; for starscapes Richardson-Lucy is always recommended.

Other PSF Generation Methods

Other PSF generation methods worthy of mention are the manual disc profile and the Airy disc. The disc profile can be used to improve images where the focus is slightly off. Try to match the size of the disc to the size of the out-of-focus blur. The Airy disc can be used to fix the slight blurring caused by the diffraction of the telescope tube itself.

Совет: If you have exceptional seeing (little to no atmospheric blur) deconvolving the image using an Airy disc may be all that you need.

Реконструируя изображение

Once you've generated a PSF you're happy with, you're ready to deconvolve your image. It is important to use the right settings to get good results.

Совет: Deconvolution is quite slow for large images. To make it quicker to find the best parameters, save your work at this point and crop a small representative part of the image. Deconvolve this with various settings, using the Undo button until you're happy. Then undo once more to get back to your un-cropped image, and apply the settings to the whole image.
Images with Stars

Images containing stars, especially linear (unstretched) data, should always be deconvolved using the Richardson-Lucy methods. Ignore Split Bregman and Wiener: those algorithms are suited to solar system images.

Deep space images pose 2 challenges with deconvolution: ringing around bright stars, and noise amplification in the background.

To deal with rings around stars, try using the Gradient Descent method and increase the number of iterations gradually until you start to see signs of dark rings forming around stars, then reduce the iterations just a little.

The above animation shows the effect of reducing the number of iterations of the multiplicative formulation of Richardson-Lucy: it also demonstrates the finer control that can be achieved by using the gradient descent method, at the cost of more iterations.

To deal with amplification of background noise, you can try applying a little noise reduction before deconvolution. In the Noise Reduction dialog, choose the Anscombe VST secondary denoising algorithm and leave the modulation quite low, try around 50-60%. You just want to take the edge off the noise to allow you to push the number of iterations a little further, not generate a completely smooth image.

Lunar Images

Typically you may wish to sharpen a lunar image after stacking. Stacked lunar images can be sharpened very nicely using the Split Bregman or Wiener methods. My usual choice is Split Bregman. Try leaving the value of $\alpha$ at the default, and deconvolving the image using a blind estimated PSF. An example of this is shown below using a freshly stacked lunar image (i.e. no wavelet processing has been done to it). Despite the limitations of the GIF animation format the sharpening can clearly be seen; it is also clear that the results from Split Bregman and Wiener are very similar.

Stacked Planetary Images

A typical planetary workflow involves stacking the planetary SER video in a specialist tool such as Autostakkert! or Astrosurface, and then sharpening the resulting image using wavelets and deconvolution. A combination of Siril’s A trous Wavelets tool and the Deconvolution tool gives excellent results as shown here. This image of Jupiter was initially sharpened using wavelets with the first layer control set to 75, the second set to 10 and the others all left at the default. A colour PSF was then built from 3 Airy discs calculated for the telescope and sensor used (a 6" Newtonian with a 3x Barlow lens and an ASI462MC sensor with 2.9 micron pixels) composited using the RGB composition tool. This was used to deconvolve the image with 6 iterations of Richardson-Lucy (here I used the multiplicative version). At each step the image becomes sharper.

Raw stack, still blurry.

Processed with Siril wavelet decomposition, wavelet layer 1 strength 75, wavelet layer 2 strength 10.

Processed with Siril wavelets as above, and then with 6 iterations of multiplicative Richardson-Lucy deconvolution.
8.3. Фильтры
Unstacked Planetary Sequences

**Совет:** Warning: this method is extremely slow as it requires individual processing of typically 30,000 (or more) images in a planetary sequence!

Some users have suggested mitigating telescope diffraction pre-stacking by deconvolving your sequence using an Airy disc PSF. To do this with a typical one-shot colour planetary camera, the sequence has to be set to debayer on load. You can take this one step further if you wish by generating three separate Airy discs for red, green and blue wavelengths (typically 600nm, 530nm and 450nm respectively). Siril cannot directly generate a colour PSF (the deconvolution UI is busy enough!) but if you save each of the red, green and blue Airy discs separately you can combine them into a colour PSF using the RGB composition tool. Save this, and if a colour or sequence is loaded the PSF will load in colour and will deconvolve each colour channel using the appropriate PSF.

Stacked and sharpened without individually deconvolving frames.

Raw stack: best 30% of 91k frames individually deconvolved using Siril.

Result of sharpening the individually deconvolved stack.
In the image above a slight improvement to the shape of the edge is evident in the version that was frame-by-frame deconvolved with an Airy disc PSF using Siril's Richardson-Lucy method prior to stacking, but care must be taken to avoid loss of details. This process is very slow: my development machine took 4.5 hours to deconvolve each of the 91k frames in this sequence, and the improvement may be minor if any.

Commands

```
makepsf clear
makepsf load filename
makepsf save [filename]
makepsf stars [-sym] [-ks=] [-savepsf=]
```

Generates a PSF for use with deconvolution. One of the following must be given as the first argument: `clear` (clears the existing PSF), `load` (loads a PSF from a file), `save` (saves the current PSF), `blind` (blind estimate of the PSF), `stars` (generates a PSF based on measured stars from the image) or `manual` (generates a PSF manually based on a function and parameters).
No additional arguments are required when using the clear argument.

To load a previously saved PSF the load argument requires the PSF filename as a second argument. This may be in any format that Siril has been compiled with support for, but it must be square and should ideally be odd.

To save a previously generated PSF the argument save is used. Optionally, a filename may be provided (this must have one of the extensions ".fit", ".fits", ".fts" or ".tif") but if none is provided the PSF will be named based on the name of the open file or sequence.

For blind, the following optional arguments may be provided: -l0 uses the l0 descent method, -si uses the spectral irregularity method, -multiscale configures the l0 method to do a multi-scale PSF estimate, -lambda= provides the regularization constant.

For PSF from detected stars the only optional parameter is -sym, which configures the PSF to be symmetric.

For a manual PSF, one of -gaussian, -moffat, -disc or -airy can be provided to specify the PSF function. Gaussian by default. For Gaussian or Moffat PSFs the optional arguments -fwhm=, -angle= and -ratio= may be provided. For Moffat PSFs the optional argument -beta= may also be provided. If these values are omitted, they default to the same values as in the deconvolution dialog. For disc PSF only the argument -fwhm= is required, which for this function is used to set the diameter of the PSF. For Airy PSFs the following arguments may be provided: -dia= (sets the telescope diameter), -fl= (sets the telescope focal length), -wl= (sets the wavelength to calculate the Airy diffraction pattern for), -pixelsize= (sets the sensor pixel size), -obstruct= (sets the central obstruction as a percentage of the overall aperture area). If these parameters are not provided, wavelength will default to 525nm and central obstruction will default to 0%. Siril will attempt to read the others from the open image, but some imaging software may not provide all of them in which case you will get bad results, and note the metadata may not be populated for SER format videos. You will learn from experience which are safe to omit for your particular imaging setup.

For any of the above PSF generation options the optional argument -ks= may be provided to set the PSF dimension, and the optional argument -savepsf=filename may be used to save the generated PSF: a filename must be provided and the same filename extension requirements apply as for makespf save filename.

Links: psf

Командная строка Siril

rl [-loadpsf=} [-alpha=} [-iters=} [-stop=} [-gdstep=} [-tv] [-fh] [-mul]

Restores an image using the Richardson-Lucy method.

Optionally, a PSF may be loaded using the argument -loadpsf=filename.

The number of iterations is provide by -iters (the default is 10).
The type of regularization can be set with `-tv` for Total Variation, or `-fh` for the Frobenius norm of the Hessian matrix (the default is none) and `-alpha=` provides the regularization strength (lower value = more regularization, default = 3000).

By default the gradient descent method is used with a default step size of 0.0005, however the multiplicative method may be specified with `-mul`.

The stopping criterion may be activated by specifying a stopping limit with `-stop=`

Links: `psf`

---

Командная строка Siril

```
sb [-loadpsf=] [-alpha=] [-iters=]
```

Restores an image using the Split Bregman method.

Optionally, a PSF may be loaded using the argument `-loadpsf=filename`.

The number of iterations is provide by `-iters` (the default is 1).

The regularization factor `-alpha=` provides the regularization strength (lower value = more regularization, default = 3000)

Links: `psf`

---

Командная строка Siril

```
wiener [-loadpsf=] [-alpha=]
```

Restores an image using the Wiener deconvolution method.

Optionally, a PSF may be loaded using the argument `-loadpsf=filename`.

The parameter `-alpha=` provides the Gaussian noise modelled regularization factor

Links: `psf`
8.3.6 Fourrier Transform

A Fourier transform (FT) is a mathematical transform that decomposes functions into frequency components, which are represented by the output of the transform as a function of frequency. This transformation is widely used in imaging because it allows to see signals at regular frequencies.

Siril allows to transform an image in the frequency space thanks to a Fast Fourier Transform algorithm. The result is in the form of two images. The first one, automatically loaded, contains the magnitude (or modulus) of the transform, the second one contains the phase. The location of the two images must be entered in the Direct Transform tab (see illustration below) of the dialog. It is then possible to modify the modulus image by removing frequency peaks corresponding to unwanted signals. It is important not to forget to save the changes.

The Centered option, when checked, centers the origin of the Direct Fourier Transform. If not, the origin is at the top-left corner.

![Fourier Transform dialog](image)

Рис. 32: Direct Transform tab.

To reconstruct the image, click on the Inverse Transform tab and enter the filepath of the modulus and phase images.

**Командная строка Siril**

```
fftd modulus phase
```

Applies a Fast Fourier Transform to the image loaded in memory. **Modulus** and **phase** given in argument are saved in FITS files.
Retrieves corrected image applying an inverse transformation. The **modulus** and **phase** used are the files given in argument.

### 8.3.7 Median Filter

The median represents the middle data point that half of data is smaller and half of data larger than this point. This is a robust estimator to remove outliers from a data set. Consequently, this tool can be useful as a naive denoiser, effective against *impulse noise*.

The layout of the window dialog is quite simple and few settings are available.

- **Kernel size**: From $3 \times 3$ to $15 \times 15$, this defines the size of a squared kernel that is used to apply the
filter. The larger the kernel, the more blurred the result will be.

- **Iterations**: This defines the number of passes of the kernel.
- **Modulation**: In Siril, modulation is a parameter between 0 and 1 mixing the original and processed images. A value of 1 keeps only the processed image, a value of 0 does not apply any median filter at all.

### Командная строка Siril

```
fmedian ksize modulation
```

Performs a median filter of size \texttt{ksize} x \texttt{ksize} (\texttt{ksize} MUST be odd) to the original image with a modulation parameter \texttt{modulation}.

The output pixel is computed as : \texttt{out}=\texttt{mod} x \texttt{m} \texttt{+ (1} mod\texttt{)} x \texttt{in}, where \texttt{m} is the median-filtered pixel value. A modulation's value of 1 will apply no modulation.

### 8.3.8 Noise Reduction

**Image Noise**

Images suffer from various types of noise:

1. **Impulse noise**
   - This type of noise (sometimes called “salt and pepper noise”) typically arises from hot or cold pixels. It is usually dealt with by using sigma rejection stacking, but sometimes you may need to deal with it if processing a single unstacked image.

2. **Additive White Gaussian Noise**
   - This type of noise is typical of well-illuminated photographs: it arises from the thermal and electronic fluctuations of the acquisition device, and the noise level is independent of the signal. AWGN can be reduced at capture time by using cooled cameras, and it is reduced in stacking because stacking \texttt{n} images increases the correlated signal by a factor of \texttt{n} whereas the uncorrelated noise only increases by a factor of \sqrt{\texttt{n}}. It is also the type of noise that most classical denoising algorithms are designed to remove.

3. **Poisson Noise**
   - When dealing with photon-starved images, the character of the noise ceases to be primarily Gaussian and the probabilistic nature of photon counting becomes significant or even dominant. This is modelled by a Poisson distribution and this type of noise is signal dependent.
Noise Reduction in Siril

Siril provides well-studied state-of-the-art classical denoising algorithms. The criteria for choosing algorithms were:

- The algorithm should be analysed in peer reviewed academic journals, with a description of the algorithm and an objective quantitative comparison of its performance.
- The authors should have made available a F/OSS implementation. This is important to avoid IP issues and, where the reference implementations have been used directly, to ensure licence compatibility.
- The algorithms should perform at a reasonable speed.
- Finally, the implementation of the algorithm must be capable of processing 32 bit floating point pixel data.

Neural network denoising technology was investigated, but discounted at the current time on the grounds of development complexity. The denoising performance of neural networks can typically beat classical approaches by up to a dB peak signal-to-noise ratio, but performance is highly dependent on the neural network being trained on data representative of the real live data.

Рис. 35: Noise Reduction dialog
Algorithms: Impulse Noise

Siril primarily removes impulse noise through sigma rejection stacking. If you use this stacking method, you shouldn’t have any issues with impulse noise. However if you are working on a single exposure you may well find impulse noise in your image. This should be dealt with using Siril’s **Cosmetic Correction** function before any other noise reduction is used, as the presence of impulse noise can skew AWGN denoising algorithms and create artefacts. It works in a similar way to sigma rejection, but on neighbouring pixels. Any pixel whose intensity is more than \( n \) standard deviations away from its neighbours will be rejected and replaced by a value based on the median of the neighbours. In the Denoising tool Cosmetic Correction is active by default and will take place before any additional denoising steps. (Even if impulse noise removal has already been carried out, leaving the setting on does no harm.) Alternatively, Cosmetic correction can be applied manually using the **Cosmetic Correction** tool in the *Image Processing* menu.

Algorithms: Additive White Gaussian Noise

The main AWGN noise reduction algorithm used in Siril is Non-Local Bayesian (NL-Bayes) denoising [Lebrun2013].

- Non-local denoising algorithms represented a major improvement over previous pixel-centred linear filters. NL-Bayes is an improved version of the earlier non-local denoising algorithms and offers one of the best classical AWGN denoising algorithms. It is marginally better than the modern “benchmark” algorithm Block Matching and 3D transform (BM3D) noise reduction and much faster to execute.

- The key parameter required to optimise the performance of AWGN algorithms is sigma, the standard deviation of the noise. Siril measures the noise level directly from the image data and passes this to the NL-Bayes algorithm, therefore in the Siril denoising tool there are no configurable inputs to NL-Bayes.

Siril complements NL-Bayes with a number of other noise reduction algorithms:

- **Data-Adaptive Dual Domain Denoising (DA3D)** [Pierazzo2017]
  - This takes the output of NL-Bayes and uses it as a guide image. This guide image is used to reprocess the original image by performing frequency domain shrinkage on shape and data-adaptive patches. It slightly improves the performance of NL-Bayes at some additional computational cost. The shape and data-adaptive patches are dynamically selected, thus concentrating the computations on the areas with greatest image detail. It can also help to reduce staircase artefacts present in the guide image.
  - In the Siril denoising tool, DA3D is a simple toggle with no optional settings.

- **Strengthen, Operate, Subtract iteration (SOS)** [Romano2015]
  - SOS works by iterating the primary denoising algorithm several times. At each iteration the image is "strengthened" by mixing in a proportion of the original noisy image. The NL-Bayes algorithm runs on this strengthened image, after which the previous estimation is subtracted.
  - The image \( x \) at an iteration \( k+1 \) is given by \( x_{k+1} = f(y + x_k) - x_k \) where \( y \) is the noisy input image.
  - In the Siril denoising tool, SOS is a toggle with two parameters: the number of iterations can be set, and the proportion of the noisy image mixed in at each iteration (rho) can be set. Avoid setting rho too high as it can result in issues with SOS converging: the default values (3 iterations and rho = 0.2) are usually fine.
Algorithms: Poisson and Poisson-Gaussian Noise

- Anscombe Variance-Stabilising Transform [Mäkitalo2011], [Mäkitalo2012]
  
  - Variance stabilising transforms are used for images with Poisson or Poisson-Gaussian noise to
    minimise the signal dependence of the noise and make it look more like AWGN, which NL-Bayes
    is good at removing, and then an inverse transform is applied on completion. The transform chosen
    for use in Siril is the Anscombe transform $A : x \rightarrow 2 \times \sqrt{(x + 3/8)}$
  
  - As the transform is non-linear, use of the direct algebraic inverse results would bias the output.
    Siril therefore uses a closed-form approximation to the exact unbiased inverse, which is quick
    to calculate and produces a substantial improvement over other forms of inverse such as the
    asymptotic inverse.
  
  - In the Siril denoising tool, the Anscombe VST is a simple toggle with no optional settings.

Note that only one of the above mentioned complementary denoising algorithms can be chosen.

The animation below shows what is possible using variance stabilisation with a photon-starved image, in
this case a single 5 minute red filter sub of the Pelican nebula, shown with the AutoStretch screen transfer
function. Note the lack of blurring, bloating or loss of detail around stars and the sharp edge of the nebula in
the bottom left part of the image compared with what might be obtained through more basic noise reduction
schemes. Once stretched more sympathetically and combined with other channels this would greatly improve
the quality achievable from very limited data (though more data is always the better solution!)

Рис. 36: Denoising a photon-starved image

Modulation

In Siril, modulation is a parameter between 0 and 1 mixing the original and denoised images. A value of
1 keeps only the denoised image, a value of 0 does not apply any denoising at all. Modulation obviously
reduces denoising performance, but in some instances if denoising has left flat areas of the image looking a
little too smooth, you can use some modulation to restore the appearance of microtexture in these regions.

When to run Noise Reduction

The noise reduction algorithms are designed to remove AWGN and should therefore perform best on
unstretched images: if white noise has a non-linear stretch applied, its characteristics change and it is
no longer white. Performing noise reduction on stretched images can still be done and will result in an
improvement, but potentially will not be as effective as if applied at the linear stage.

Noise Reduction Interface

The Siril Noise Reduction tool can be accessed in two ways: via the GUI, or via the command interface. The
GUI is shown below. Note: the SOS advanced options are hidden if SOS is not selected.

Noise reduction can also be applied using Siril commands, either in the console or in scripts. The format is:

Командная строка Siril

```
denoise [-nocosmetic] [-mod=m] [ -vst | -da3d | -sos=n [-rho=r] ] [-indep]
```
Рис. 37: Siril Noise Reduction GUI
Denoises the image using the non-local Bayesian algorithm described by Lebrun, Buades and Morel.

It is strongly recommended to apply cosmetic correction to remove salt and pepper noise before running denoise, and by default this command will apply cosmetic correction automatically. However, if this has already been carried out earlier in the workflow it may be disabled here using the optional command `-nocosmetic`.

An optional argument `-mod=m` may be given, where $0 <= m <= 1$. The output pixel is computed as: $out = m \times d + (1 - m) \times in$, where $d$ is the denoised pixel value. A modulation value of 1 will apply no modulation. If the parameter is omitted, it defaults to 1.

The optional argument `-vst` can be used to apply the generalised Anscombe variance stabilising transform prior to NL-Bayes. This is useful with photon-starved images such as single subs, where the noise follows a Poisson or Poisson-Gaussian distribution rather than being primarily Gaussian. It cannot be used in conjunction with DA3D or SOS, and for denoising stacked images it is usually not beneficial.

The optional argument `-da3d` can be used to enable Data-Adaptive Dual Domain Denoising (DA3D) as a final stage denoising algorithm. This uses the output of BM3D as a guide image to refine the denoising. It improves detail and reduces staircasing artefacts.

The optional argument `-sos=n` can be used to enable Strengthen-Operate-Subtract (SOS) iterative denoise boosting, with the number of iterations specified by $n$. In particular, this booster may produce better results if the un-boosted NL-Bayes algorithm produces artefacts in background areas. If both `-da3d` and `-sos=n` are specified, the last to be specified will apply.

The optional argument `-rho=r` may be specified, where $0 < r < 1$. This is used by the SOS booster to determine the amount of noisy image added in to the intermediate result between each iteration. If `-sos=n` is not specified then the parameter is ignored.

The default is not to apply DA3D or SOS, as the improvement in denoising is usually relatively small and these techniques requires additional processing time.

In very rare cases, blocky coloured artefacts may be found in the output when denoising colour images. The optional argument `-indep` can be used to prevent this by denoising each channel separately. This is slower but will eliminate artefacts.

**Comparison**

The images below provide a simplistic comparison of the different algorithms. Note that only one image is used: in practice, different algorithms will be better suited for use on different images. All the images can be clicked on to view at 100% zoom.
Original noisy image

Рис. 38: Noisy image

Denoised with NL-Bayes only

Denoised with NL-Bayes only, with 75% modulation to restore some microtexture

Denoised with NL-Bayes using the Anscombe transform

Denoised with DA3D using a NL-Bayes guide image

Denoised with NL-Bayes and SOS

Limitations

The main limitation is that the algorithms work best when the noise is Gaussian in character (or can be made approximately Gaussian using the VST). There are some reasons why this might not be true:

- If the image has already been heavily processed, for example with deconvolution or wavelet sharpening, the character of the noise will not generally be Gaussian any more. If both noise reduction and deconvolution form part of your workflow, noise reduction should be done first.

- OSC images may denoise less well than mono or composited colour images. A small reduction of luminance AWGN is achieved but as a result of the deBayering process the character of the noise is changed so that it is no longer well modelled as AWGN, and is not removed very effectively. Additionally, for both OSC and composited mono colour images, chrominance noise tends not to be well modelled as AWGN and requires different treatment. At present chrominance noise is best tackled in general purpose image manipulation software such as The GIMP.
Рис. 39: Denoised with NL-Bayes only

Рис. 40: Use of modulation
Рис. 41: Denoised with NL-Bayes, variance stabilised with Anscombe transform. A 200% uninterpolated zoom is shown to the right.

Рис. 42: Denoised with DA3D, guide image prepared using NL-Bayes. A 200% uninterpolated zoom is shown to the right.
Рис. 43: Denoised with NL-Bayes and SOS iterations. A 200% uninterpolated zoom is shown to the right.

Литература

8.3.9 Rotational Gradient (Larson Sekanina Filter)

The rotational gradient, also called Larson Sekanina filter, is a filter that allows to remove circular structures from an image, to better highlight other details. This technique is particularly effective to show the jets coming out of the nucleus of a comet.

The principle is quite simple: this image processing consists in subtracting two copies of the image from each other, one of the two copies having been previously rotated with respect to a point defined in the image.

- If there are circular structures around this point they are not modified by rotation and will disappear after rotation.
- If there are non-circular structures, like jets in the coma, they will be shifted in relation to each other between the two copies and the subtraction will amplify the contrast of this structure in the result.
- If the comet moves in the image, it is possible to add a radial shift.

In the example below, concerning the comet 46-P Wirtanen, the alignment was made on the comet and the stars show important trails. The comet is very circular and it is difficult to see details about its activity. Therefore, it is not necessary to define a radial shift. For the rotation, an angle of a little more than 28° was chosen (this choice was made after several attempts and using the undo button to go back). To choose the coordinates of the center of rotation, just make a selection around the cometary nucleus and click on **Use current selection**. This action will copy the coordinates of the centroid to the desired location.

A simple click on **Apply** will apply the filter. In our example, the tail becomes visible.

Командная строка Siril

8.3. Фильтры
8.4 Star Processing

Stars are an integral part of deep sky images and play a crucial role in bringing out the beauty and detail of celestial objects. They often appear as brilliant dots of light, showcasing their brightness and colors, making deep sky images truly captivating. However, due to the limitations of observing conditions, the stars in these images may appear larger and over-exposed. To combat this, astronomers use advanced image processing techniques to separately process the stars and control their size and brightness in the final image.

This part of the documentation is then dedicated to everything related to the stars processing.
Рис. 45: Image of a comet whose tail is barely visible.
Рис. 46: After applying the filter, the tail of the comet appears very clearly.
8.4.1 StarNet Star Removal

StarNet is a software developed by Nikita Misiura. Its first version was released under a free and opensource license. Unfortunately, version 2 became proprietary and the sources are since closed. The version 2 is available free of charge from there. Make sure you download the Command Line Tool version. Siril can interface with any version of the StarNet CLI tool, including the new experimental Torch-based version that has initially been released for M1- and M2- based Apple Macs.

**Warning:** On MacOS, for Siril to detect and use StarNet correctly, it is necessary to fix some permissions and security issues first. Start by opening the Terminal application from the Utilities folder within Applications. In Terminal, you need to change your working directory from your home directory to the StarNetCLI installation directory. To do this type in `cd` followed by a space and then drag the StarNetCLI folder into the terminal window to copy its path. Press `enter`. Then type in the following four commands, pressing `enter` after each one:

```bash
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
```

Then, at the first use with Siril, the execution of StarNet may fail with a warning about libtensorflow. Cancel out of this warning. Open System Preferences and under Privacy and Security click the **Allow anyways** button for libtensorflow. After this, StarNet should execute properly in Siril.

**Warning:** On MacOS, again, there is a new Starnet executable optimized for the Apple Silicon chip that has been released on the site: https://www.starnetastro.com/experimental/. This new version is much faster than previous version because it uses the new MPS accelerated PyTorch (https://developer.apple.com/metal/pytorch/). Also, this version contains signed binaries, follow the installation instructions in the README.txt

However, it is still possible for Siril to run external binaries and this is what we decided to implement starting with Siril 1.2.0. For the settings, please refer to the preference page. It explains how to tell Siril where StarNet is located.

**Warning:** This is the location of the command line version of StarNet that need to be given, not the GUI one.

Note that StarNet requires its input in the form of TIFF images, therefore if Siril is compiled without libtiff support then the StarNet integration will not be available.

The primary purpose of StarNet is to remove all the stars from the images in order to apply a different process between the stars and the rest of the image. This usually helps to control star bloat during the different stretches, but it is also very useful for creating images of comets where the comet tracking rate can be significantly different to the distant stars.

The tool is very easy to use and only five options are available:

- **Pre-stretch linear image**: If selected, an optimised Midtone Transfer Function (MTF) stretch is applied to the image before running StarNet, and the inverse stretch is applied on completion. This is necessary for using StarNet at the linear stage of processing.

- **Recompose stars on completion**: If selected, on completion of the star removal process the star
recomposition tool will open, providing an interface for independently stretching and blending the background and the stars if star reduction, rather than total removal, is desired. This option has no effect when processing a sequence.

- **Generate star mask**: This will generate a star mask and saved it in the working directory. The star mask is calculated as the difference between the original image and the starless image. The default is to produce a star mask.

- **Upsample x2**: This option will up-sample the image by a factor of 2 before running StarNet. This improves performance on very tight stars but quadruples processing time and may impair performance on very large stars. The image is rescaled to the original size on completion.

- **Use custom stride**: A custom value may be entered for the StarNet stride parameter. The default value is 256 and the StarNet developer recommends not to change this.

The StarNet process can easily be applied to a sequence. The togglebutton *Apply to sequence* selects whether the process will apply to a single image or to a sequence. Where the process is applied to a sequence, a new sequence will be created containing the starless images and, if star mask generation is selected, a second sequence will be created containing the corresponding star mask images.

More information about StarNet can be found on the original website.

A click on *Execute* will run the process. It can be slow, depending of your machine performance. However, Siril shows a progressbar to follow the processing. As with other Siril processes, if processing a sequence the progressbar will only update after completion of each image in the sequence, and will show the overall progress through the sequence.
**Prerequisite:** StarNet is an external program, with no affiliation with Siril, and must be installed correctly prior to first use of this command, with the path to its installation directory correctly set in Preferences / Miscellaneous. The directory must contain the Command Line Tool version (not GUI version which exists for Windows users).

The starless image is loaded on completion, and a star mask image is created in the working directory unless the optional parameter `-nostarmask` is provided.

Optionally, parameters may be passed to the command:
- 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.

More tips and tricks are available [there](#).

### Командная строка Siril

```bash
seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

This command calls StarNet++ to remove stars from the current sequence. The first argument must be the sequence from which to remove stars: all the other available arguments are the same as for the STARNET command.

Links: [starnet](#)

### 8.4.2 Восстановить звёзды

Star Recomposition is a GUI tool to aid in combining starless and star mask images. It doesn't provide any unique image manipulation that can't be done in other ways, for example using PixelMath and the Generalized Hyperbolic Stretch tool, but it does provide a real-time preview of the combination of two separate images with different stretches applied to each.

There is no command-line equivalent for this tool as it is purely graphical in nature, however starless and star mask images could be combined using the `pm` and GHT-related commands (`ght`, `invght`, `modasinh`, `invmodasinh` and `linstretch`).

The tool is found in the Image Processing menu, in the Star Processing sub-menu.

The dialog is divided into two columns, one for each of the input images.

Each input image is loaded using the respective file chooser. Each column has a stretch histogram preview, which may be minimized to aid use on small displays, a set of GHS stretch controls, and Reset and Apply buttons.
Рис. 48: Star Recomposition dialog box.
The histogram mode can be changed between linear and logarithmic using the toggle at the bottom of the dialog. This dialog obeys the Siril-wide preference for linear or logarithmic histograms that can be set in the Preferences window.

**Simple Mode**

The dialog has two views, which determines what controls are shown. It opens in Simple mode, which shows only the most useful controls for a typical starless / starmask combination.

- The stretch type for the starless image is set to Generalized Hyperbolic stretch and the Stretch Factor, Local stretch intensity, Symmetry Point and Black Point controls are shown. As well as using the SP control, the Symmetry Point can be set using the eyedropper tool to select the average pixel value of a selection from the image. *Note that the eyedropper tool is disabled when there is an unapplied BP shift: because of the process of applying the hyperbolic stretch and then the BP shift, the behaviour of the tool becomes non-intuitive when a non-zero BP parameter is set. To resolve this, simply apply the BP shift and the eyedropper will become available again for your next hyperbolic stretch.*

- The stretch type for the star mask image is set to Modified Arcsinh stretch and the Stretch Factor and Highlight Protection controls are shown.

- The human-weighted luminance color model is used for both sets of stretches: this does a better job of preserving colors in the unstretched image.

Details of all the stretch controls, both those shown in Simple mode as well as those shown in Advanced mode, can be found on the Generalized Hyperbolic Stretch documentation page.

The BP control works in a slightly different way to the BP control in the standalone GHS linear (black point adjust) stretch. In this tool the Black Point adjustment is applied after the hyperbolic stretch, whereas in the standalone tool it is a separate stretch applied by itself. When trying to optimize the combination of independent stretches to the two input images, this was found to be the most workable approach. It does mean that the amount of black point shift required in this tool is different to the amount required in the GHS tool, and that the Black Point cannot be set by clicking on the histogram.

Each stretch is independent. The stretch settings for the starless side can be applied using the left-hand Apply button: this stretches the starless image according to the current stretch settings and then resets the stretch settings so that further stretches can be applied in an iterative manner. Similarly, the stretch settings for the star mask can be applied using the right-hand Apply button. Either set of stretch settings can be reset to the defaults using its respective Reset button.

The dialog can be toggled between Simple and Advanced mode using the button at the bottom.

**Advanced Mode**

In Advanced mode the full range of GHS stretch controls are available including Stretch Type, Colour stretch model and Shadow protection point for both input images. This allows greater customization of the two stretches if required. If the user interface is put back to Simple mode, any changes made using the advanced controls remain in effect, only the controls are hidden.

**Примечание:** It is not possible to stretch the saturation channels in this tool. The tool is already quite memory-hungry and CPU intensive: doubling the memory requirement by adding a HSL copy of each working image is considered excessive. Saturation can easily be stretched separately after the combination is complete.
8.4.3 Desaturate Stars

When a star finder is applied to an image (whose data are always linear), ellipses are displayed around the stars. When an ellipse is magenta, it means that the star is saturated.

A saturated star is a star whose brightest pixels have no more information and are clipped to the maximum value. In general we try to not to saturate the stars, even if this is not possible for the brightest. If despite the precautions there are still saturated stars, Siril has an algorithm that will reconstruct the profile of the star taking into account the results of the adjustment made during the findstar.

First, you need to perform a star detection, either with the \texttt{findstar} command or the button of the \textit{Dynamic PSF} Window. Then, the desaturate tool is found in \textit{Star Processing $\rightarrow$ Desaturate Stars}.

Совет: We recommend using a Moffat profile in the \textit{Dynamic PSF} window to get better parameters.

| Предупреждение: | It is important to run this tool on linear images, otherwise the stars will not have a Gaussian/Moffat profile and the calculations will be invalid. |

After clicking on the tool, Siril switches to the console and displays the results of the current process:

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

(continues on next page)
Рис. 50: A star detection shows all stars found by Siril. Magenta ellipses are for saturated stars. The image is displayed in autostretch view: data are still linear.
22:26:21: Remapping output to floating point range 0.0 to 1.0
22:26:21: Execution time: 4.09 s

It is necessary to run a star detection again to see the changes.

Рис. 51: After a desaturate processing, no more magenta ellipses are visible. All stars have been reconstructed. The image is displayed in autostretch view: data are still linear.

**Командная строка Siril**

```
unclipstars
```

Re-profiles clipped stars to desaturate them, scaling the output so that all pixel values are $\leq 1.0$
8.4.4 Полный повторный синтез

The Full Resynthesis tool aims to help to fix badly distorted stars using Siril's star fitting functions. It can be helpful for rescuing images that suffer from bad coma or other distortions. If Siril can detect the stars it can fix them.

The tool is located in the Image Processing menu, in the Star Processing sub-menu.

The output of the tool is a synthetic star mask. In order to make use of this, it must be recombined with a starless version of the original image. This can be prepared using the starnet command or Starnet GUI tool, or using third party star removal software.

This tool has no options, you simply click on the menu item to use it, or use the command synthstar.

If no stars have been detected in the image, the tool will automatically detect stars using the current star modelling parameters accessible via the Dynamic PSF tool or using the setfindstar command.

If stars have been modelled using the Dynamic PSF tool or the findstar command, the detected stars will be resynthesized using their individual modelled luminosity profiles. A shortcut to the Dynamic PSF tool is provided by means of the configuration button in the GUI menu next to the Full Resynthesis tool.

It is recommended to carry out star detection manually first, as it allows verification of the results: if any galaxies have been incorrectly detected as stars, they can be removed from the list of stars before running resynthesis.

Once the synthetic star mask has been created it can be combined with the starless image using the Star Recombination tool.
Команды

Командная строка Siril

**synthstar**

Synthstar fixes bad stars. No matter how much coma, tracking drift or other distortion your stars have, if Siril's star finder routine can detect it, synthstar will fix it. To use intensive care, you may wish to manually detect all the stars you wish to fix. This can be done using the findstar console command or the Dynamic PSF dialog. If you have not run star detection, it will be run automatically with default settings.

For best results synthstar should be run before stretching.

The output of synthstar is a fully corrected synthetic star mask comprising perfectly round star PSFs (Moffat or Gaussian profiles depending on star saturation) computed to match the intensity, FWHM, hue and saturation measured for each star detected in the input image. This can then be recombined with the starless image to produce an image with perfect stars.

No parameters are required for this command

Links: *psf*

---

**8.5 Geometry**

**8.5.1 Rotate**

**Rotate 90 degrees**

It is possible to rotate the image 90 degrees clockwise and counterclockwise with the dedicated menu. Here the rotation is done without interpolation of the pixels and it is therefore the preferred method if you want to rotate the image by a multiple of 90 degrees. This feature is also reachable through the icons in the toolbar.
**Rotate&Crop**

For a rotation of another angle you have to use the Rotate&Crop tool. It allows a precise rotation and cropping that can be easily controlled.

![Rotate&Crop dialog box](image)

Рис. 53: Rotate&Crop dialog box showing all settings.

Five interpolation algorithms are available:
- Ближайший сосед
- Билинейная
- Бикубическая
- Отношение площади пикселя
- Lanczos-4 (Default)

Lanczos-4 is the one that gives the best results. However, if you see artifacts, especially stars surrounded by black pixels, then you may want to try others. However, the button **Interpolation clamping** applies a clamping factor to Bicubic and Lanczos-4 interpolation in order to prevent ringing artifacts.

If you don't want the image to be cropped after rotation, then you should uncheck the **crop** button. However, the missing areas of the picture will be filled with black pixel.

The interest of this tool is that the rotation of the image is represented by a red frame, as illustrated in the figure below. In addition, if a selection is active, it is possible to change its size and see in real time the framing evolve.

**Командная строка Siril**

```
rotatePi
```

Вращает изображение на 180° вокруг центра. Это эквивалентно команде "ROTATE 180" или "ROTATE -180"

Links: rotate
Рис. 54: Rotate&Crop dialog box with a active selection. Click to enlarge the figure and see the details better.
Rotates the image by an angle of degree value. The option -nocrop can be added to avoid cropping to the image size (black borders will be added).

Note: if a selection is active, i.e. by using a command `boxselect` before `rotate`, the resulting image will be a rotated crop. In this particular case, the option -nocrop will be ignored if passed.

The pixel interpolation method can be specified with the -interp= argument followed by one of the methods in the list no[ne], ne[arest], cu[bic], la[nzos4], li[near], ar[ea]. If none is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the -noclamp argument.

### 8.5.2 Mirror

It is also possible to apply a mirror transformation to the image. Either along the x axis or along the y axis.

This transformation is also accessible via the buttons \[ \text{R} \] and \[ \text{R} \] of the toolbar.

Flips the image about the horizontal axis. Option -bottomup will only flip it if it’s not already bottom-up.
8.5.3 Биннинг

The binning is a special transformation for resampling image. It computes the sum or mean of the pixels 2x2, 3x3, ... (depending of the binning factor) of the in-memory image (like the analogic binning of CCD camera).

Рис. 55: Binning dialog box

Командная строка Siril

```bash
binxy coefficient [-sum]
```

Computes the numerical binning of the in-memory image (sum of the pixels 2x2, 3x3..., like the analogic binning of CCD camera). If the optional argument `-sum` is passed, then the sum of pixels is computed, while it is the average when no optional argument is provided.

8.5.4 Масштабировать

The resample tool allows to resize the image at the cost of an interpolation chosen from the following list:

- Ближайший сосед
- Билинейная
- Бикубическая
- Отношение площади пикселя
- Lanczos-4 (Default)

Lanczos-4 is the one that gives the best results. However, if you see artifacts, especially stars surrounded by black pixels, then you may want to try others. However, the button `Interpolation clamping` applies a clamping factor to Bicubic and Lanczos-4 interpolation in order to prevent ringing artifacts.

If you want to change the image ratio, then you should uncheck the `Preserve Aspect Ratio` button.
Командная строка Siril

```bash
resample { factor | -width= | -height= } [-interp=] [-noclamp]
```

Размножение изображения, либо с фактором `factor`, либо для цели размера вдоль `width` или `height` предоставлена либо в `width=`, либо в `height=`. Это, в общем случае, используется для изменения размера изображений, фактор 0.5 разделяет размер вдвое.

В графическом пользовательском интерфейсе, мы можем видеть, что несколько алгоритмов интерполяции предложены.

Пиксельная интерполяция метод может быть указан с помощью аргумента `-interp=` с одним из методов в списке `no`, `ne`, `bic`, `la`, `lncos4`, `li`, `near`, `ar`}. Если `none` не указан, преобразование вынуждается сдвинуть и пиксельная сдвига будет применена к каждому изображению без использования интерполяции.

Клининг биксицику и ланцос4 интерполяции методов является предпочтительным, чтобы избежать артефактов, но может быть отключен с помощью `-noclamp` аргумент.

8.6 Извлечение фона

Фон неба часто имеет нежелательный градиент, вызванный световым загрязнением, Луной, или просто ориентацией камеры по отношению к земле. Эта функция получает образцы фона из большого количества участков изображения, ищет тенденцию в изменении фона и удаляет его, следуя сглаживающей функции, чтобы избежать удаления галактик вместе с фоном.

Образцы фона могут быть выбраны автоматически с заданной плотностью (Плотность образцов), которые размещаются на изображении после нажатия кнопки Создать. Если какие-то области изображения ярче медианы на произведение коэффициента допуска (Допуск) и стандартного отклонения (σ), то в этой области образец фона не размещается. После автоматического создания, образцы также могут быть добавлены (левый клик) или удалены (правый клик) вручную.
Рис. 57: Диалоговое окно "Извлечение фона". Слева версия с полиномиальной, а справа — с радиально-базисной (RBF) функциями.
Доступны два метода удаления градиента:

- **Радиально-базисная функция (RBF):** Это наиболее современный метод. Он использует Радиально-базисную функцию для создания фона неба, чтобы удалить градиент с большой гибкостью. Последняя может быть изменена с помощью параметра Сглаживание.

- **Полиномиальная функция:** Первонаучальный алгоритм, очень практичный, при удалении градиента на отдельных снимках (субэкспозициях). Для пользователей доступны четыре степени функции, от 1 до 4. Чем больше степень, тем более гибкая коррекция. Однако слишком высокая степень функции может привести к странным результатам, как перекоррекция.

Так же доступны следующие настройки:

- **Подмешать шум:** Используйте эту опцию, когда после удаления фона появляется вертикальное полосение. Этот намеренно привнесённый шум, используется для рандомизации ошибок квантования, предотвращая появление крупномасштабных градиентов на изображениях, таких как цветовое полосение.

- **Коррекция:**
  - Вычитание: главным образом корректируются аддитивные эффекты, вызванные световыми загрязнениями или Луной.
  - Деление: главным образом используется для коррекции мультипликативных эффектов, как виньетирование или дифференциальное поглощение атмосферой, например. Однако такие операции должны корректироваться с помощью мастер-кадров плоского поля.

- **Вычислить фон:** Будет рассчитан синтетический фон и применён выбранный тип коррекции. Модель всегда рассчитывается для загруженного в память изображения, позволяет пользователю работать итеративно.

- **Показать оригинальное изображение:** Удерживайте нажатой эту кнопку, чтобы увидеть оригинальное изображение.

Изображения после предобработки могут иметь комплексный градиент фона, поскольку в ходе съёмки возможно вращение градиента. Полностью удалить градиент может быть нелегко, поскольку трудно представить его через полиномиальную функцию. В таком случае стоит рассмотреть возможность удаления градиента на субэкспозициях: на отдельном изображении градиент фона гораздо проще и обычно следует простой линейной функции (степень 1).

**Командная строка Siril**

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5]
```

Вычисляет синтетический градиент фона, используя либо полиномиальную функцию степени `degree` или радиально-базисную функцию (RBF), если указан аргумент `-rbf`, и вычитает градиент из изображения.

Количество и плотность образцов в горизонтальной строке и допуск для исключения ярких областей могут быть указаны опционально с помощью соответствующих аргументов. Допуск указывается в единицах медианного абсолютного отклонения (mad): медиана + допуск * mad.

Шум, необходимый для низких динамических градиентов, может быть подмешан с помощью аргумента `-dither`.

Для RBF также доступен дополнительный параметр сглаживания.
Командная строка Siril

seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-prefix=]

Аналогична команде SUBSKY, но применяется последовательности **название** последовательности.

Dithering, required for low dynamic gradients, can be disabled with **-nodither**.

The output sequence name starts with the prefix "bkg_" unless otherwise specified with **-prefix=** option.

Ссылка: subsky

8.7 Extraction

8.7.1 Split Channels

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) see , or CIELAB.

Рис. 58: Split channels dialog box.

Совет: If no name is given to a channel, then the channel is not extracted.
Командная строка Siril

```
split file1 file2 file3 [-hsl | -hsv | -lab]
```

Splits the color image into three distinct files (one for each color) and save them in `file1.fit`, `file2.fit` and `file3.fit` files. A last argument can optionally be supplied, `-hsl`, `-hsv` or `lab` to perform an HSL, HSV or CieLAB extraction. If no option are provided, the extraction is of RGB type.

### 8.7.2 Split CFA Channels

CFA means color filter array. This term is often used to describe one-channel image content of a color image, with each pixel corresponding to values acquired behind an on-sensor filter. This is to oppose to debayer images (or debayered or demosaiced).

Opening a CFA image in Siril is required for pre-processing, like removing the dark signal before interpolating the image into 3-channel color. We can also use the color filter information to extract images like this:

- **Split CFA Channels**: four images are created from the CFA image, each representing one filter of the Bayer matrix, so in general R.fit, G1.fit, G2.fit and B.fit. It is useful if the goal is to process separately the different colors of the image.

- **Extract Ha**: using an H-alpha filter with a color camera image (OSC: on-sensor color, or one-shot color camera) means that only the pixels with red filters will be useful, so in general only a quarter of them. This function creates a new image that contains only the pixels associated with the red filter documented in the Bayer matrix of the image.

- **Extract Ha/OIII**: for OSC cameras, filters that let through photons from H-alpha and O-III wavelength have appeared. This extraction creates two images: an image from the red pixels like the Extract Ha, and an image combining the green and blue pixels into one for O-III. Both images are half the definition of the input image.

**Примечание**: The Ha/OIII resampling option is how to handle the output of Extract Ha/OIII. No resampling produces full resolution OIII image and a half resolution Ha image; upsample Ha upsamples the Ha image by a factor of 2 to match the OIII image; downsample OIII downsamples the OIII image by a factor of 2 to match the Ha image.

- **Extract Green**: for photometry, it’s often useful to only process the green part of the CFA image, because it is more sensitive and has two pixels to average, reducing noise even more. Of course, the created image also sees its definition halved by two.

**Примечание**: These functions only work if the Bayer matrix has been properly documented by the acquisition software and if the image format supports it, so in general FITS or SER.

**Предупреждение**: This does not work with other filter matrices than the Bayer matrices, like the Fujifilm X-TRANS.
8.7.3 Wavelet Layers

This tool extracts the different planes of the image by applying the wavelet process. Each plane is saved in an image and the set of images can be read as a sequence. You can choose up to 9 layers for the wavelet calculation and the type of the algorithm is either Linear or BSpline. The latter is usually the preferred one.

The decomposition is done through a number of detail layers defined at increasing characteristic scales and a final residual layer, which contains the remaining unresolved structures.
Рис. 61: Original image of M45 (courtesy of V. Cohas).
Рис. 62: 6 extracted planes.
8.8 Linear matching

Linear matching is the process of finding a linear function that matches best (in the sense of Least Squares) the intensity of pixels from one image to those of a reference image. This is a quick and easy way to balance the histograms of different images.

The **Reference** allows you to pick the reference image.

The **Reject low** and **Reject high** sliders allow to exclude pixels values in the left and right tails of the intensities distributions. They are defined as quantiles, in the range \([0, 1]\). For instance, default for high is 0.92, meaning that the 8% brightest pixels will be excluded from the fitting to find the linear match coefficients.

**Warning:** The image and reference must be aligned prior to applying a linear match. Otherwise, there is no reason to assume that their pixels intensities are correlated.

**Command line Siril**

```
linear_match reference low high
```

Computes a linear function between a **reference** image and the image in memory.

The function is then applied to the current image to match it to the reference one. The algorithm will ignore all reference pixels whose values are outside of the \([\text{low}, \text{high}]\) range.
8.9 RGB compositing

The RGB composition tool allows you to assemble up to 8 monochrome images to form a single color image. The images can be shifted by translation but not by rotation, otherwise it will not be possible to register them. In such a situation, it is necessary to create a mini sequence of the input images and register them with the global registration algorithm.

The operation of this tool is quite simple, just load the images and assign them a color. The first field, optional, is reserved for the luminance layer. Once a luminance layer is loaded you can integrate it or not in the composition thanks to the Use Luminance button. Each color can be customized by clicking on it and choosing a new one. When more than 3 images (or 4 if there is luminance) are loaded, it may be necessary to adjust the brightness of each channel. The Adjust layers brightness button performs this operation automatically.

Примечание: For binning and image dimensions, the first loaded image determines the size of the output image. If you have images of different sizes, you should always load the largest first. If your images are different just because of binning, so with the same field of view, the composition tool will upscale the smaller images when they are loaded to match the size of the first loaded image. It is useful for the common L-RGB taken with the colour filters in bin 2. This also means that if two images have not been taken with the same sensor, it is unlikely they will have the same field of view and pixel sampling after image resampling, and
this will not work with this tool.

Three color spaces are available for rendering the composition:

- **HSL** (for hue, saturation, lightness)
- **HSV** (for hue, saturation, value; also known as HSB, for hue, saturation, brightness)
- **CIE L*a*b***

and are left to the choice of the user.

Once the composition is finished, it is possible to do the color balance by clicking on the button **Finalize color balance**: this opens the **color calibration dialog**.

If the images are not aligned with each other, and they are just shifted by translation, then it is possible to align them. Two algorithms are possible:

- **One star registration** (**deep-sky**): you have to draw a selection around a star, making sure that the selection contains the star in all channels.
- **Image pattern alignment** (**planetary/deepsky**): 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.

Командная строка Siril

```
rgbcomp red green blue [-out=result_filename]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename]
```

Создаёт композитное RGB изображение, используя три независимых изображения, или LRGB композитное изображение, используя дополнительное яркостное изображение и три монохромных изображения или цветное изображение. Итоговое изображение будет иметь название composed_rgb.fit или composed_lrgb.fit, если в необязательном аргументе не указано другое название

### 8.10 Merge CFA Channels

The purpose of this tool is to combine multiple monochrome images that have been previously extracted from a CFA sensor (with the **Extraction → Split CFA channels**... menu for example). The tool merges the separate red, green (x2), and blue channel images into a single composite image called CFA image.

**Предупреждение:** This tool is dedicated to images from a Bayer matrix and therefore it cannot work with images from X-Trans files from Fuji cameras.

The dialog is split in three different parts:

- **Input files**: Select the image containing the CFA0, CFA1, CFA2 and CFA3 Bayer subpatterns. If this has been produced using Siril’s Split CFA function it will have the CFA prefix.
- **Bayer Pattern**: Sets the Bayer pattern header to be applied to the result. This must match the Bayer pattern of the image that the original Bayer subchannels were split from.
The sequence part, at the bottom, allows to process whole sequences by reconstituting a CFA image sequence. Clicking on the Apply to sequence button displays a help text to proceed correctly. This text is reported in the next tooltip. There are two available options:

- **Sequence input marker**: Identifier prefix used to denote CFA number in the separate CFA channel images. This should be set to whatever sequence prefix was used when the split_cfa process was run (default: CFA_).

- **Sequence output prefix**: Prefix of the image names resulting from the merge CFA process. By default it is mCFA_.

**Совет**: You must have the CFA0 sequence selected in the main window sequence tab.

Your separate sub-CFA sequences must have been processed in exactly the same way.

The filenames **must** be in the same directory and **must** differ only by the name of the CFA channel, i.e. if a CFA0 image is r_pp_CFA_0_Light_0001.fit, the corresponding images for the other CFA channels must be r_pp_CFA_1_Light_0001.fit, r_pp_CFA_2_Light_0001.fit and r_pp_CFA_3_Light_0001.fit.

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 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.
Командная строка Siril

`merge_cfa file_CFA0 file_CFA1 file_CFA2 file_CFA3 bayerpattern`

Builds a Bayer masked colour image from 4 separate images containing the data from Bayer subchannels CFA0, CFA1, CFA2 and CFA3. (The corresponding command to split the CFA pattern into subchannels is `split_cfa`.) This function can be used as part of a workflow applying some processing to the individual Bayer subchannels prior to demosaicing. The fifth parameter `bayerpattern` specifies the Bayer matrix pattern to recreate: `bayerpattern` should be one of 'RGGB', 'BGGR', 'GRBG' or 'GBRG'.

Командная строка Siril

`seqmerge_cfa sequencename bayerpattern [-prefixin=] [-prefixout=]`

Same command as MERGE_CFA but for the sequence `sequencename`. The Bayer pattern to be reconstructed must be provided as the second argument as one of RGGB, BGGR, GBRG or GRBG.

The input filenames contain the identifying prefix "CFA_" and a number unless otherwise specified with `-prefixin=` option.

Note: all 4 sets of input files must be present and must be consistently named, the only difference being the number after the identifying prefix.

The output sequence name starts with the prefix "mCFA_" and a number unless otherwise specified with `-prefixout=` option.

Links: `merge_cfa`

8.11 Pixel Math

One of the most powerful tools in Siril is the Pixel Math. It allows you to manipulate the pixels of the images using mathematical functions. From simple addition or subtraction, to more advanced functions, like MTF, Pixel Math is a perfect tool for astronomical image processing.

This page aims to describe the tool entirely, to see detailed examples, please refer to the excellent tutorial on the site.

The window is divided into 5 parts.
Рис. 64: Pixel Math dialog box as shown at opening
1. The first one, including 3 text zones receiving the mathematical formulas. Only the first one is used if you want to produce a monochrome image. Uncheck the *Use single RGB/K expression* button to produce RGB output.

2. The second is the variables area with the selection of *Functions* and *Operators*. Each variable is an image that must be loaded beforehand with the + button. You can click on the desired function and/or operator to make it appear in the formula entry to make it appear in the formula entry.

3. The third, the *parameters* field, allows the user to define parameters that are separated by ,. For example, if you set parameters with the expression `factor=0.8, K=0.2`, all the occurrences of *factor* and *K* in the formula above will be replaced by 0.8 and 0.2 respectively. \( \text{Ha} \cdot \text{factor} + \text{OIII} \cdot K \) would therefore evaluate to \( \text{Ha} \cdot 0.8 + \text{OIII} \cdot 0.2 \).

4. The *output* field is reserved for scaling the image within a given range. One need to expand the frame before using it.

5. Finally, the *presets* area allows the user to reuse previously saved formulas with the button to the right of the formula areas. One need to expand the frame before using it. Double-click on the formula to copy it to the right entry.
8.11.1 Usage

Name of variables

By default it is possible to load 10 images simultaneously. Each image is given a variable name starting with I followed by a number from 1 to 10. However, if the loaded image contains the keyword FILTER, then the value of the latter becomes the default variable name. Of course it is always possible to change it by double clicking on it.

![Variable table example](image)

Рис. 67: It is possible to change the name of the variable.

Examples

Let's take a monochrome image of galaxies. This is a linear data seen through the autostretch view.

![Original image](image)

Рис. 68: Original image.

The following expression:
\texttt{iif(Image>med(Image)+3*noise(Image), 1, 0)}

will produce a star mask.

Рис. 69: After the formula above.

\begin{verbatim}
Komandnaya strочка Siril

pm "expression" [-rescale [low] [high]]
\end{verbatim}

This command evaluates the expression given in argument as in PixelMath tool. The full expression must be between double quotes and variables (that are image names, without extension, located in the working directory in that case) must be surrounded by the token $, e.g. "$image1$ * 0.5 + $image2$ * 0.5". A maximum of 10 images can be used in the expression. Image can be rescaled with the option -rescale followed by \textbf{low} and \textbf{high} values in the range [0, 1]. If no low and high values are provided, default values are set to 0 and 1.
8.11.2 Functions

There are two types of functions. Those that apply directly to the pixels and those that apply to the entire image (such as the statistics functions).

<table>
<thead>
<tr>
<th>Function</th>
<th>Use case</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs</td>
<td>abs (x)</td>
<td>Absolute value of x.</td>
</tr>
<tr>
<td>acos</td>
<td>acos (x)</td>
<td>Арксосинус x.</td>
</tr>
<tr>
<td>acosh</td>
<td>acosh (x)</td>
<td>Гиперболический арксинус x.</td>
</tr>
<tr>
<td>asin</td>
<td>asin (x)</td>
<td>Арксинус x.</td>
</tr>
<tr>
<td>asinh</td>
<td>asinh (x)</td>
<td>Гиперболический арксинус x.</td>
</tr>
<tr>
<td>atan</td>
<td>atan (x)</td>
<td>Арктангенс x.</td>
</tr>
<tr>
<td>atan2</td>
<td>atan2 (y, x)</td>
<td>Арктангенс y/x.</td>
</tr>
<tr>
<td>atanh</td>
<td>atanh (x)</td>
<td>Гиперболический арктангенс x.</td>
</tr>
<tr>
<td>ceil</td>
<td>ceil (x)</td>
<td>Round x upwards to the nearest integer.</td>
</tr>
<tr>
<td>cos</td>
<td>cos (x)</td>
<td>Cosine of x.</td>
</tr>
<tr>
<td>cosh</td>
<td>cosh (x)</td>
<td>Гиперболический косинус x.</td>
</tr>
<tr>
<td>exp</td>
<td>exp (x)</td>
<td>Показательная функция.</td>
</tr>
<tr>
<td>fac</td>
<td>fac(x)</td>
<td>Факториал.</td>
</tr>
<tr>
<td>iif</td>
<td>iif(cond, expr_true, expr_false)</td>
<td>Conditional function (or inline if function). Returns expr_true if cond evaluates to nonzero. Returns expr_false if cond evaluates to zero.</td>
</tr>
<tr>
<td>floor</td>
<td>floor (x)</td>
<td>Наибольшее целое значение меньше или равное x.</td>
</tr>
<tr>
<td>log</td>
<td>log (x)</td>
<td>Natural logarithm of x.</td>
</tr>
<tr>
<td>log10</td>
<td>log10 (x)</td>
<td>Base-10 logarithm of x.</td>
</tr>
<tr>
<td>log2</td>
<td>log2 (x)</td>
<td>Base-2 logarithm of x.</td>
</tr>
<tr>
<td>max</td>
<td>max (x, y)</td>
<td>Функция максимума.</td>
</tr>
<tr>
<td>min</td>
<td>min (x, y)</td>
<td>Функция максимума.</td>
</tr>
<tr>
<td>mtf</td>
<td>mtf (m, x)</td>
<td>Передаточная функция полутонов (MTF) x для параметра баланса полутонов m в диапазоне [0, 1].</td>
</tr>
<tr>
<td>ncr</td>
<td>ncr (x, y)</td>
<td>Функция сочетаний.</td>
</tr>
<tr>
<td>npr</td>
<td>npr (x, y)</td>
<td>Функция перестановок.</td>
</tr>
<tr>
<td>pi</td>
<td>pi</td>
<td>The constant π=3.141592...</td>
</tr>
<tr>
<td>pow</td>
<td>pow(x, y)</td>
<td>Exponentiation function.</td>
</tr>
<tr>
<td>sign</td>
<td>sign(x)</td>
<td>Sign of x:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>+1 if x &gt; 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 if x &lt; 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 if x = 0.</td>
</tr>
<tr>
<td>sin</td>
<td>sin (x)</td>
<td>Sine of x.</td>
</tr>
<tr>
<td>sinh</td>
<td>sinh (x)</td>
<td>Гиперболический синус x.</td>
</tr>
<tr>
<td>sqrt</td>
<td>sqrt (x)</td>
<td>Square root of x.</td>
</tr>
<tr>
<td>tan</td>
<td>tan (x)</td>
<td>Tangent of x.</td>
</tr>
<tr>
<td>tanh</td>
<td>tanh (x)</td>
<td>Гиперболический тангенс x.</td>
</tr>
<tr>
<td>trunc</td>
<td>trunc (x)</td>
<td>Усечённая целая часть x.</td>
</tr>
<tr>
<td>Function</td>
<td>Use case</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>adev</td>
<td>adev (Image)</td>
<td>Среднее абсолютное отклонение изображения.</td>
</tr>
<tr>
<td>bwmv</td>
<td>bwmv (Image)</td>
<td>Двухвесовая средняя дисперсия изображения.</td>
</tr>
<tr>
<td>height</td>
<td>height (Image)</td>
<td>Height in pixels of the specified image.</td>
</tr>
<tr>
<td>mad</td>
<td>mad (Image)</td>
<td>Median absolute deviation of the image. The use of mdev is also possible.</td>
</tr>
<tr>
<td>max</td>
<td>max (Image)</td>
<td>Максимальное значение пикселей изображения.</td>
</tr>
<tr>
<td>mean</td>
<td>mean (Image)</td>
<td>Mean of the image.</td>
</tr>
<tr>
<td>med</td>
<td>med (Image)</td>
<td>Median of the image. The use of median is also possible.</td>
</tr>
<tr>
<td>min</td>
<td>min (Image)</td>
<td>Минимальное значение пикселей изображения.</td>
</tr>
<tr>
<td>noise</td>
<td>noise (Image)</td>
<td>Estimation of Gaussian noise in the image.</td>
</tr>
<tr>
<td>sdev</td>
<td>sdev (Image)</td>
<td>Standard deviation of the image.</td>
</tr>
<tr>
<td>width</td>
<td>width (Image)</td>
<td>Width in pixels of the specified image.</td>
</tr>
</tbody>
</table>

### 8.11.3 Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Use case</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>~</td>
<td>~x</td>
<td>Оператор инверсии пикселя.</td>
</tr>
<tr>
<td>-</td>
<td>-x</td>
<td>Унарный оператор минус (смена знака).</td>
</tr>
<tr>
<td>+</td>
<td>+x</td>
<td>Унарный оператор плюс.</td>
</tr>
<tr>
<td>!</td>
<td>!x</td>
<td>Logical NOT operator.</td>
</tr>
<tr>
<td>^</td>
<td>x ^ y</td>
<td>Exponentiation operator.</td>
</tr>
<tr>
<td>*</td>
<td>x * y</td>
<td>Multiplication operator.</td>
</tr>
<tr>
<td>/</td>
<td>x / y</td>
<td>Division operator.</td>
</tr>
<tr>
<td>\</td>
<td>x % y</td>
<td>Оператор модуля.</td>
</tr>
<tr>
<td>+</td>
<td>x + y</td>
<td>Addition operator.</td>
</tr>
<tr>
<td>-</td>
<td>x - y</td>
<td>Subtraction operator.</td>
</tr>
<tr>
<td>&lt;</td>
<td>x &lt; y</td>
<td>Оператор сравнения МЕНЬШЕ.</td>
</tr>
<tr>
<td>&lt;=</td>
<td>x &lt;= y</td>
<td>Less Than Or Equal relational operator.</td>
</tr>
<tr>
<td>&gt;</td>
<td>x &gt; y</td>
<td>Оператор сравнения БОЛЬШЕ.</td>
</tr>
<tr>
<td>&gt;=</td>
<td>x &gt;= y</td>
<td>Greater Than Or Equal relational operator.</td>
</tr>
<tr>
<td>==</td>
<td>x == y</td>
<td>Оператор сравнения РАВЕН.</td>
</tr>
<tr>
<td>!=</td>
<td>x != y</td>
<td>Not Equal To relational operator.</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>x &amp;&amp; y</td>
<td>Логический оператор И.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
В Siril есть вкладка, отображающая графики на основе данных, полученных в ходе выравнивания или других вычислений. Эта вкладка очень мощная и позволяет легко сортировать изображения и выполнять их анализ. На вкладку График можно перейти с помощью клавиатурной комбинации F5.

9.1 Регистрация

Для улучшения ручной сортировки кадров после регистрации, на вкладку График были добавлены возможности для построения графиков. По окончании регистрации последовательности (или при загрузке последовательности, имеющей данные регистрации) с помощью выпадающего списка возможно указать параметры для построения графиков и сортировки имеющихся данных.

Во время сортировки каждая новая последовательность вставляется в окончательный список с отсортированными кадрами. Материалы из каждого кадра накапливаются и затем сортируются в порядке возрастания. Если для построения графиков выбран параметр Кол-во звёзд, построение графиков происходит с использованием вариации Качество относительно опорного изображения, полученную от последовательности с минимальным значением Кол-во звёзд.

Вы можете также построить график одного параметра по отношению к другому. В выпадающем списке доступны:

- **FWHM**: Полная ширина на уровне половины высоты (полуширина), один из наиболее частых критериев для оценки изображения глубокого неба.
- **Округлость**: Округлость $r$ вычисляется как соотношение $\frac{FWHM_y}{FWHM_x}$.
- **wFWHM**: Это улучшение простой FWHM. Позволяет исключить гораздо менее информативные изображения, сравнивая количество обнаруженных звёзд с таковым на опорном изображении.
- **Фон**: Среднее значение фона неба.
- **Кол-во звёзд**: Количество звёзд, использованных для регистрации.
- **Позиция по X**: сдвиг по оси X относительно опорного изображения.
- **Позиция по Y**: сдвиг по оси Y относительно опорного изображения.
- **Качество**: Это число в диапазоне $[0, 1]$, которое определяет качество изображений, обработанных любым алгоритмом регистрации планет.
Рис. 1: Вкладка График после завершения регистрации всего звёздного неба.
Рис. 2: The values of roundness vs FWHM are displayed as a scatter plot. Hover onto the different data points to show X and Y values, together with the corresponding frame number.
Рис. 3: Different possibility of graphics taken with the same set of images.
Click on one of the data point to exclude the frame or to open it. The later option will load the picture and pop out the frame selector. The parameter chosen for Y values is reflected in the frame selector last column, which can then be used to sort, review and select or unselect subs from the sequence.

![Frame Selector](image.png)

Рис. 4: Right-click on a data point to exclude or to load it in the Image preview

You can also mass select/unselect multiple data points by drawing a selection onto the plot. The information at the top of the selection tells the number of points selected, as well as the boundary values of your selection. You can reshape the selection just as you would do with a drawn selection in the Image view. Once happy with your selection, a right-click will display a menu to keep or exclude the points, or to set the zoom to the selection.

### 9.2 Фотометрия

To complement sorting and filtering frames from the sequence, you can also perform a PSF on a star for the full sequence. The procedure is detailed in the [photometry](#) page. Then the photometry item in the first dropdown list becomes sensitive and is automatically selected. The other dropdown list contain the following items:

- **FWHM**: Maximum width at half maximum, as defined above.
- **Округлость**: Округлостью \( r \) вычисляется как соотношение \( \frac{FWHM_y}{FWHM_x} \).
- **Amplitude**: It is the maximum value of the fitted function, located at the centroid coordinates.
- **Magnitude**: Relative magnitude of the analyzed star.
- **Background**: Average of the local sky background value taken in the annulus.
- **Позиция по X**: сдвиг по оси X относительно опорного изображения.
- **Позиция по Y**: сдвиг по оси Y относительно опорного изображения.
- **SNR**: An estimator of the signal-to-noise ratio.

In photometry, unlike registration, it is not possible to change the X axis. And only the number of images can be used (or the Julian day).

- The *Clear all* and *Clear latest* buttons allow you to erase either all the photometric curves or the last one drawn.
- More information about *NINA exoplanet* can be found in the *Light Curve* page.
Рис. 5: Right-click on a drawn selection to mass select/unselect or to zoom
9.3 Shared options

- Each plot can be saved in png format by clicking on the Save Plot button. The file, whose name includes a timestamp, is saved in the working directory.
- The Export to CSV button exports the displayed plot to a CSV file.
- If the sampling of the images is known, then the arcsec button displays the FWHM in arcsec instead of pixel.

9.4 Plot interactions

Here is a summary of the possible interactions with the plot window:

- **Left-click** in a slider: puts the nearest red dot on it
- **Double-click** in a slider: resets this axis
- **Right-click + drag** in a slider: move the zoom on this axis
- **Left-click + drag** in the plot area: draws a selection
- **Left-click + drag** on a selection edge: resize the selection
- **Double-click** in the plot area: reset the zoom on the 2 axes
- **Right-click** when a selection is active: display the menu to: zoom on the selection/ keep only the points of the selection/ exclude the points of the selection
- **Left-click** when a selection is active: delete the selection

*Совет:* Наведите курсор на слово "легенда", и появится описание легенды графика.

- Фиолетовая кривая: фактический график согласно селекторам X и Y.
- Зелёная кривая: отсортированные значения в порядке уменьшения качества.
- Окружность: значение опорного кадра.
- Крест: значение загруженного в настоящий момент кадра.
Глава 9. Построение графиков
Глава 10

Динамическая PSF

Данный раздел описывает два важных шага, выполняемых для определения звёзд на отдельных кадрах. Определение на отдельном изображении можно выполнить или уточнить с помощью Основное меню → Данные об изображении → Динамическая PSF или с помощью клавиатурной комбинации Ctrl + F6.

Рис. 1: Динамическая PSF, выполняемая на изображении глубокого космоса.

Обработка выполняется следующим образом:

- первоначальное определение кандидатов в звёзды
- подгонка модели PSF для каждого кандидата. Проверка на адекватность, основываясь на параметрах подогнанной модели, чтобы убедиться, что кандидат является звездой и отбросить кандидаты, не являющиеся звёздами.
Результатом этого процесса является список звёзд, с их положением на изображении относительно левого верхнего угла и измеренными величинами всех звёзд в списке.

10.1 Первоначальные кандидаты в звёзды

Хотя при взгляде на изображение кажется очевидным, где располагаются звёзды, это несколько более трудная задача формализовать этот процесс в математических терминах и критериях. В этом разделе коротко описан алгоритм, лежащий в основе определения звёзд. Он вдохновлен руководством к программе DAOPHOT [Stetson1987], с некоторыми упрощениями, улучшающими производительность. Оригинальный алгоритм был нацелен на поиск всех возможных звёзд, служа целями создания каталогов звёзд. Siril же нуждается в определении звёзд как деталей для регистрации. Он также должен уметь обрабатывать разнообразные изображения, которые предоставляют пользователи — большинству из нас недоступно профессиональное астрономическое оборудование на своём заднем дворе, и нам приходится делать определённый выбор, касающийся предварительных знаний об условиях получения изображений (масштаб, условия съёмки и пр.).

С течением времени, наша реализация алгоритма эволюционировала к тому, что она представляет собой сейчас. Она имеет целью не пропустить слишком яркие звёзды, которые важны для регистрации и выбраковать как можно больше выбросов, делая это с разумной скоростью.

Его можно разделить на следующие шаги:

- вычисляется статистика изображения для получения уровня фона, в виде медианы изображения, и его шума. Это предполагает, что изображение относительно плоское. Как следствие, определение имеет тенденцию быть менее эффективным в углах, если после калибровки осталось сильное виньетирование.

- Так же вычисляется динамический диапазон, определяемый как разница между максимумом изображения и его фоном. Позже это будет полезно для определения насыщенных звёзд.

- изображение сглаживается при помощи гауссовского ядра. Для идеального сглаживания было бы необходимо использовать ядро того же размера, как FWHM изображения. Вместо этого, мы выбрали произвольный размер, который даёт удовлетворительный результат в очень большом диапазоне условий. Это позволяет быть "слепым" к условиям получения изображения.

- на сглаженном изображении, обнаруживаются локальные максимы выше уровня фона плюс X раз его шума (X может быть изменено с помощью параметра threshold в интерфейсе). Проверьте, чтобы это было максимумом в определённом размере (задаваемом radius параметром).

- выполните тест стабильности, чтобы убедиться, что максимум и его соседи гораздо выше окружающих пикселей (например, отвергайте участки в ярких участках межзвёздной среды).

- выполните тест стабильности, чтобы предположить, что ядро вокруг максимума является насыщенным, т. е. постоянно близко к верхней границе диапазона. Если это так, выполните алгоритм обхода края, чтобы определить предел насыщенного участка.

- используйте первую и вторую производные по горизонтальной и вертикальной линии, проходящей через центр, чтобы предположить локальный фон звезды, амплитуду и ширину в всех направлениях (вверх, вниз, влево и вправо).

- Если параметры симметричны в всех направлениях (до параметра roundness), подтвердите звезду как потенциальный кандидат.

Одна из причин, по которой этот алгоритм был выбран, это его простота и гибкость. Он может работать с любыми изображениями, которые пользователи могут предоставить, а также может быть настроена на различные условия получения изображений. Это позволяет быть "слепым" к условиям получения изображения.
10.2 Models

Two models are used in the Dynamic PSF window. In general, Moffat is much better suited to fit objects such as stars.

1. An elliptical Gaussian fitting function defined as

\[
G(x, y) = B + Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)}.
\]  

2. An elliptical Moffat PSF fitting function defined as

\[
M(x, y) = B + A \left(1 + \left(\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)^{-\beta}\right),
\]  

where:

- \( B \) is the average local background.
- \( A \) is the amplitude, which is the maximum value of the fitted PSF.
- \( x_0, y_0 \) are the centroid coordinates in pixel units.
- \( \sigma_x, \sigma_y \) are the standard deviation of the Gaussian distribution on the horizontal and vertical axes, measured in pixels.
- \( \beta \) is the exponent from the Moffat formula that controls the overall shape of the fitting function. The upper bound of this parameter was set to 10. A higher value is meaningless and means that the Gaussian is good enough to fit the star.

Other parameters are derived from these fitted variables:

- FWHM_x and FWHM_y: The Full Width Half Maximum on the X and Y axis in pixel units. These parameters are calculated as follow:
  - FWHM_x = \( 2\sigma_x\sqrt{2\log 2} \).
  - FWHM_y = \( 2\sigma_y\sqrt{2\log 2} \).
- It is possible to obtain the FWHM parameters in arcseconds units. This requires you fill all fields corresponding to your camera and lens/telescope focal in the setting parameter window in the burger menu, then Image Information and Information. If standard FITS keywords FOCALLEN, XPIXSZ, YPIXSZ, XBINNING and YBINNING are read in the FITS HDU, the PSF will also compute the image scale in arcseconds per pixel.
- \( r \): The roundness parameter. It is expressed as FWHM_x/FWHM_y, with FWHM_x > FWHM_y the symmetry condition.
- Another parameters is also fitted in both Gaussian and Moffat models. This is the rotation angle \( \theta \), defined in the range \([-90, +90]\). The addition of this parameter implies a coordinate change where the \( x \) and \( y \) variables expressed in (10.1) and (10.2) are replaced by \( x' \) and \( y' \):

\[
\begin{align*}
x' &= +x \cos \theta + y \sin \theta \\
y' &= -x \sin \theta + y \cos \theta.
\end{align*}
\]
Рис. 2: Displays of two circular PSFs according to a Gaussian profile and a Moffat profile. Both models use the same parameters and the Moffat profile uses $\beta = 1.4$. 
Рис. 3: Rotated Gaussian and Moffat function have $\sigma_x = 2\sigma_y$, $\theta = 45$. For Moffat, $\beta = 1.4$. 
Рис. 4: Star Profile with Gaussian and Moffat models. Several $\beta$ values are tried. $\beta = 10$ gives a profile very close to the Gaussian one.
10.3 Minimization

Minimization is performed with a non-linear Levenberg-Marquardt algorithm thanks to the very robust GNU Scientific Library. This algorithm is used to find the minimum of a function that maps a set of parameters to a set of observed values. It is a combination of two optimization techniques: the gradient descent method and the inverse-Hessian method.

The Levenberg-Marquardt algorithm adjusts the trade-off between these two methods based on the curvature of the function being minimized. When the curvature is small, the algorithm uses the gradient descent method, and when the curvature is large, the algorithm uses the inverse-Hessian method.

Since version 1.2.0, the saturated part of the star is removed from the fitting process, enabling to capture much more accurately the non-saturated part. This is what enables to "reconstruct" the star profile when using the Desaturate menu option or unclipstars command.

10.4 Use

Dynamic PSF can be called from two different ways depending on what you want:

You may want to fit just one or a few stars. In this case, after drawing a selection around an unsaturated star (this is important for the accuracy of the result) you can either right click and choose the Pick A Star item, click the + button in the Dynamic PSF dialog, or type Ctrl + Space. An ellipse is then drawn around the star. To open the dialog it is also possible to use the shortcut Ctrl + F6.

You may want to analyze as many stars as possible by clicking on the icon , or using the command line findstar. All detected stars are surrounded by an ellipse: orange if the star is ok, magenta if the star is saturated. It is also possible to show the average of the computed parameters as illustrated below, by clicking on the button.

Star detection has a number of uses:

- Siril uses it internally for astrometric purposes when registering sequences of images. This is automatic and requires no user intervention.

- Because stars are so bright compared with dim features of interest such as nebulae or galaxies, it is very common that some stars in an image will be saturated, meaning their brightness profile is clipped. This can cause problems for some image processing functions, particularly deconvolution, and results in loss of colour information and slightly greater star bloat when applying stretches. Analysis of all the stars will show you which ones are saturated, and you can then use the Desaturate menu option or unclipstars command to fix the problem by synthesis of the clipped part of the profile.

Command line Siril

unclipstars

Re-profiles clipped stars to desaturate them, scaling the output so that all pixel values are \(<= 1.0\)

- Ideally all stars in an image should be perfectly round, however problems such as coma, astigmatism and bad tracking, as well as issues such as incorrect back focus to field flatteners, can give rise to
Рис. 5: Average of the fitted stars in the Gaussian model.
patterns of elliptical stars in an image. The ellipses produced by the Dynamic PSF tool can give a good visual illustration of such problems.

- Examination of the average star parameters, especially FWHM and the Moffat fitting function beta parameter, can provide information about the quality of seeing in an image.
- Detection of all stars is the first step in using the Star Tools → Full Resynthesis tool. This synthesizes corrected luminosity profiles for all detected stars, and can be used to create a synthetic star mask which can then be mixed with a starless image generated by Starnet++ to fix otherwise irredeemable stars in an image. In this case detection of stars using the Moffat profile may give a more realistic result and can also make it easier to filter out galaxies incorrectly detected as stars by using the Minimum beta setting.
- The Center Selected Star toggle button can be used to find a particular star in the list quickly and easily in the image, by centring it in the viewport. This is useful if you have detected all stars and wish to check specific solutions to ensure they really are a star and not a galaxy or a cosmic ray.
- Similarly to this, clicking on an orange or magenta star ellipse in the main viewport will highlight the selected star solution in the Dynamic PSF dialog. This could be useful if you wish to see the parameters of an individual star.
- Future developments in Siril's deconvolution functions may support using Dynamic PSF measurements to synthesize a deconvolution function that matches star parameters measured directly from the image.

### 10.5 Configuration

Dynamic PSF can be configured using the settings in the Dynamic PSF dialog:

- **Radius** sets the half-size of the search box. If you have trouble detecting particular stars you can try changing this but usually the default is fine.
- **Threshold** changes the threshold above the noise for star detection. If you increase this value, fewer faint stars will be detected. You may still wish to do this for very noisy images though. Decreasing this value may detect more faint stars, but will also make the algorithm more likely to misidentify random noise spikes as stars.
- **Roundness** threshold sets the allowable ellipticity for detections to be accepted as stars. Highly elliptical stars may occur due to imperfect tracking or aberrations, but sometimes stars that are too close from each others are also detected as a single very elongated star. To highlight all these problems, it is possible to enable a higher bound for the roundness. A maximum value of 1 is equivalent to disabling the range, leaving only the minimum value. This roundness range should be disabled for registration or astrometry.
- **Convergence** sets a criterion used by the solver. Increasing it will allow the solver more iterations to converge and can potentially detect additional stars, but may increase the solver running time.
- **Profile type** chooses between solving Gaussian or Moffat profiles for the stars.
- **Minimum beta** sets a minimum permissible value of beta for a detection to be accepted as a star. Galaxies may sometimes be detected as Moffat profile stars but they have diffuse profiles and the value of beta is usually very low, less than around 1.5.
- **Relax PSF checks** allows for relaxation of several of the star candidate quality checks. This is likely to result in a significant increase in false positive star detections, often with wild parameters.
- A range of **minimum** and **maximum amplitude** can be set to limit the amplitude (parameter named $A$ in reports) of detected stars. This is useful if only non-saturated stars need to be selected, for PSF fitting in deconvolution for example. Note that removing the saturated stars from detection can break registration and astrometry.
The settings defined in this window can be tested on the currently loaded image. However, you have to keep in mind that they will also be used for all images of the sequence, especially for the global alignment registration.

The `findstar` command will obey the same settings entered in the Dynamic PSF dialog, but it can also be configured using the `setfindstar` command.

**Командная строка Siril**

```
findstar [-out=] [-layer=] [-maxstars=]
```

Detects stars in the currently loaded image, having a level greater than a threshold computed by Siril. After that, a PSF is applied and Siril rejects all detected structures that don't fulfill a set of prescribed detection criteria, that can be tuned with command SETFINDSTAR. Finally, a circle is drawn around detected stars.

Optional parameter `-out=` enables to save the results to the given path.
Option `-layer=` specifies the layer onto which the detection is performed (for color images only). You can also limit the max number of stars detected by passing a value to option `-maxstars=`.

См. также команду CLEARSTAR

Links: psf, setfindstar, clearstar

**Командная строка Siril**

```
```

Defines stars detection parameters for FINDSTAR and REGISTER commands.

Passing no parameter lists the current values.
Passing `reset` resets all values to defaults. You can then still pass values after this keyword.

Configurable values:

- `-radius=` defines the radius of the initial search box and must be between 3 and 50.
- `-sigma=` defines the threshold above noise and must be greater or equal to 0.05.
- `-roundness=` defines minimum star roundness and must between 0 and 0.95. `-maxR` allows an upper bound to roundness to be set, to visualize only the areas where stars are significantly elongated, do not change for registration.
-minA and -maxA define limits for the minimum and maximum amplitude of stars to keep, normalized between 0 and 1.

-focal= defines the focal length of the telescope.

-pixelsize= defines the pixel size of the sensor.

-gaussian and -moffat configure the solver model to be used (Gaussian is the default).
If Moffat is selected, -minbeta= defines the minimum value of beta for which candidate stars will be accepted and must be greater or equal to 0.0 and less than 10.0.

-convergence= defines the number of iterations performed to fit PSF and should be set between 1 and 3 (more tolerant).

-relax= relaxes the checks that are done on star candidates to assess if they are stars or not, to allow objects not shaped like stars to still be accepted (off by default)

Links: findstar, register, psf

10.6 Литература
Astrometry is the science dealing with the positions and motions of celestial objects. Astrometry is essential in modern astrophotography where capture software like N.I.N.A, Ekos, APT or others, plate solve the images in order to obtain an astrometric solution, meaning they will precisely know the position of the frame with regards to the sky. Astrometry can also be used at processing stage, like in photometric color calibration tool for example.

### 11.1 Platesolving

The platesolving is a major step in astronomical image processing. It allows images to be associated with celestial coordinates, giving the ability to know what object is in the observed field of view. Many of Siril's tools, such as the Photometric Color Calibration (PCC) tool, need to know the coordinates of the image with sufficient accuracy in order to work.

Astrometry in Siril can be performed in a few different ways:

- Using the dedicated tool accessible via the burger menu → Image Information → Image Plate Solver, or using the shortcut Ctrl + Shift + A.
- Using the photometric color calibration tool, based on the same tool but extended to add star color analysis and comparison with star colors in catalogs to adjust the image's color, available in the Image Processing menu → Color Calibration → Photometric Color Calibration or using the shortcut Ctrl + Shift + P.
- Using the platesolve command, introduced in Siril 1.2.

Since version 1.2, plate solving can be done by two different algorithms. The first was the only one in Siril until this version, it's based on the global registration's star matching algorithm, trying to register images onto a virtual image of a catalog with the same field of view. The second is new, it is using an external program called solve-field from the Astrometry.net suite, installed locally. For Windows platforms, the simplest way to get it is to use ansvr.

Astrometric solutions require a few parameters to be found, like image sampling. The window of the tool helps gathering those parameters, we will now see how to fill them correctly.
Рис. 1: Platesolving dialog
Рис. 2: Инструмент фотометрической калибровки цвета
11.1.1 Image parameters

Target coordinates

Finding an astrometric solution is easier and faster when we roughly know where we are looking. Siril’s plate solver, as it’s comparing a catalog with the image, needs to know approximately the position of the center of the image to get the catalog excerpt. Astrometry.net has all the catalogs it needs locally, so it can browse through all of it to find a solution, but it is of course much faster to tell it where to start.

Acquisition software often also control the telescope nowadays and should know the approximate coordinates where the image was taken. In that case, using a FITS format, these coordinates will be provided in the image metadata, the FITS header. This is not always the case, and clearly not the case when RAW DSLR images are created instead of FITS.

When opening the plate solver or PCC windows, the current image's metadata is loaded and displayed in the window. If no coordinates appear at the top, or if RA and Dec remain zero, some user input is needed. If you don’t know at all what the image is, use a blind solve with astrometry.net. Otherwise, provide equatorial J2000 coordinates corresponding to as close as the center of the image as possible, either by filling the fields if you already know the coordinates, or by making a query with an object name (not yet possible from the command).

The text field at the top left of the window is the search field, pressing Enter or clicking the Find button will make a Web request to convert the object name to coordinates. Several results may be found with the entered name, they will be displayed in the list below. Selecting one updates the coordinates at the top.

It is also possible to choose the server on which you want to execute the query, it does not change the results much, but sometimes one of them can be online, so others would act as a backup, between CDS, VizieR and SIMBAD (default).

Примечание: If the object is not found, please change the name you enter: you need to use the name written in the astronomical catalogue. For example, for the Bubble Nebula, please enter NGC 7635 and not Bubble Nebula.

The coordinate fields are filled in automatically, but it is possible to define your own. Don’t forget to check the S box if the object you are looking for is located in the southern hemisphere of the sky (negative declinations).

Image sampling

Image sampling is the most important parameter for plate solving. Given in arcseconds per pixel in our case, it represents how much zoomed on the sky the image is, so how wide a field to search for.

It is derived from two parameters: focal length and pixel size. They are often available in the image metadata as well. When not available from the image, the values stored in the settings are used. The values of the images and of the preferences can be set using the Information dialog. In any case, check the displayed value before plate solving and correct if needed. If an astrometric solution is found, the default focal length and pixel size will be overwritten. This behavior can be disabled in the settings.

Предупреждение: If binning was used, it should be specified in the FITS header, but this can take two forms: the pixel size can remain the same and the binning multiplier should be used to compute the sampling, or the pixel size has already multiplied by the acquisition software. Depending on the case you are facing, either of the forms can be chosen from the preferences or from the Information window.
Pixel size is given in the specification of astronomical cameras, and can generally be found on the Web for DSLR or other cameras. The number of sensors is limited and most of them are known.

Focal length depends on the main instrument, but also on backfocus and correcting or zooming lenses used. Give a value as close as what you believe the effective focal to be, if an astrometric solution is found, the computed focal length will be given in the results and you will be able to reuse that in your acquisition software and for future uses of the tool.

When either of the fields is updated, the sampling is recomputed and displayed in the window (called 'resolution' here). Make sure the value is as close as reality as possible.

**Other parameters**

Finally, there are three toggle buttons at the bottom of the frame:

1. The option *Downsample image* downsamples the input image to speed-up star detection in it. The downside is that it may not find enough stars or give a less accurate astrometric solution. The size of the output image remains unchanged.

2. If the image is detected as being upside-down by the astrometric solution, with the option *Flip image if needed* enabled, it will be flipped at the end. This can be useful depending on the capture software, if the image has not the right orientation when it is displayed in Siril (see more explanations).

3. When the option *Auto-crop (for wide field)* is applied, it performs a platesolve only in the center of the image. This is only done with wide field images (larger than 5 degrees) where distortions away from the center are important enough to fool the tool. Ignored for astrometry.net solves.

### 11.1.2 Catalogue parameters

By default, this section is insensitive because everything is set to automatic. By unchecking the auto box, however, it is possible to choose the online catalog used for the platesolve, which may depend on the resolution of the image. The choice is done between:

- **TYCHO2**, a catalogue containing positions, proper motions, and two-color photometric data for 2,539,913 of the brightest stars in the Milky Way.
- **NOMAD**, a simple merge of data from the Hipparcos, Tycho-2, UCAC2, Yellow-Blue 6, and USNO-B catalogs for astrometry and optical photometry, supplemented by 2MASS near-infrared. The almost 100 GB dataset contains astrometric and photometric data for about 1.1 billion stars.
- **Gaia DR3**, released on 13 June 2022. The five-parameter astrometric solution, positions on the sky (, ), parallaxes, and proper motions, are given for around 1.46 billion sources, with a limiting magnitude of G = 21.
- **PPMXL**, a catalog of positions, proper motions, 2MASS- and optical photometry of 900 million stars and galaxies.
- **Bright Stars**, a star catalogue that lists all stars of stellar magnitude 6.5 or brighter, which is roughly every star visible to the naked eye from Earth. The catalog contains 9,110 objects.

Примечание: An internet connection is required to use these online catalogs.

The *Catalogue Limit Mag* is an option that allows you to limit the magnitude of the stars retrieved in the catalog. The automatic value is calculated from the image resolution.
Using local catalogues

With version 1.1, starting in June of 2022, it was possible to rely on a locally installed star catalogue, for disconnected or more resilient operation. The star catalogue we found to be the most adapted to our needs is the one from KStars. It is in fact composed of four catalogues (documented here in KStars), two of them not being directly distributed in the base KStars installation files:

- **namedstars.dat**, the brightest stars, all of them have names
- **unnamedstars.dat**, also bright stars, but down to magnitude 8
- **deepstars.dat**, fainter stars extracted from The Tycho-2 Catalogue of the 2.5 Million Brightest Stars, down to magnitude 12.5
- **USNO-NOMAD-1e8.dat**, an extract of the huge NOMAD catalogue limited to B-V photometric information and star proper motion in a compact binary form, down to magnitude 18.

When comparing these catalogues with the online NOMAD, we can easily see that many stars are missing. If not enough are found for your narrow field, you should still use the remote queries. A nice thing to check when the catalogues are installed is highlighting which stars of the image will be used for the PCC, those available with photometric information in the catalogues, using the `nomad` command.

Download

The first two files are available in KStars source, the Tycho-2 catalogue from a debian package and the NOMAD catalogue from KStars files too, as documented in this small article for KStars installation. It is has multiple worldwide mirrors as indicated in the articles.

To make things easier to Siril users, and possibly to KStars users too, we redistribute the four files in a single place, and in a more compressed format. With the LZMA algorithm (used by xz or 7zip), the file size is 1.0GB instead of the 1.4GB with the original gzip file.

To make it available from anywhere faster, it is distributed with bittorrent, using this torrent file or the following magnet link.

Slower direct download links are available here (right click on each file name on the left and save the links).

Установка в Siril

The files can be put anywhere and their paths given to Siril in the settings, but there is a default location for the four files: `~/.local/share/kstars/` on Linux. They can be linked there to avoid unnecessary copies. Settings can be changed from the command line now, using the `set` command.

When available and readable, Siril will not use the Web service to retrieve astrometric or photometric data. See the messages in the log tab or on the console to verify that the catalogue files are used as expected.

Only SIMBAD will be used to convert object names into coordinates if required, but that should only be needed if the acquisition software did not record the target coordinates in the FITS header, or when using SER file format which cannot hold this information.
**Usage**

With the addition of the new link between Siril’s plate solver and the local catalogue and the new link between Siril’s PCC and the local catalogue, a new command nomad was created to display which stars in a plate solved image contain photometric information (the B-V index) and can be used for calibration.

This is a good way to verify that the plate solving and the image are aligned, in addition to the object annotation feature (see *annotations*).
Technical information

For photometry, Siril only uses the B-V index, which gives information about star colour. The three image channels are then scaled to give the best colour representation to all stars in the image.

For more information about the KStar binary file type, see this page and this discussion on kstars-devel and some development notes in Siril here and here.

Sha1 sums for the 4 catalogue files:

<table>
<thead>
<tr>
<th>Sha1 Sum</th>
<th>Catalogue File</th>
</tr>
</thead>
<tbody>
<tr>
<td>4642698f4b7b5ea3bd3e9edc7c4df2e6ce9c9f7d</td>
<td>namedstars.dat</td>
</tr>
<tr>
<td>53a336a41f0f3949120e9662a465b60160c9d0f7</td>
<td>unnamedstars.dat</td>
</tr>
<tr>
<td>d32b78fd6a3f977fa853d299c44ee0014c2ab53</td>
<td>deepstars.dat</td>
</tr>
<tr>
<td>12e663e04cae9e43fc4de62d6eb2c69905ea513f</td>
<td>USNO-NOMAD-1e8.dat</td>
</tr>
</tbody>
</table>

Licenses for the 4 catalogue files.

11.1.3 Using the local astrometry.net solver

Since version 1.2, solve-field, the solver from the astrometry.net suite, can be used by Siril to plate solve images or sequence of images.

For Windows platforms, the simplest way to get it is to use ansvr. If you did not modify the default installation directory, that is \%LOCALAPPDATA\%\cygwin_ansvr, Siril will search for it without additional setup. If you have cygwin and have build astrometry.net from the sources, you must specify the location of cygwin root in the Preferences.

For other OS, the executable is expected to be found in the PATH.

The use of this tool makes it possible to blindly solve images, without a priori knowledge of the area of the sky they contain. It’s also a good alternative to Siril’s plate solver in case it fails, because it’s a dedicated and proven tool that also can take field distortion into account.

Default settings should be fine, but can be modified if you really want to, using the set command (default values specified between parens) or in the Astrometry tab of preferences. How wide the range of allowed scales is (15%), how big the radius of the search from initial coordinates is (10 degrees), the polynomial order for field distortion (0, disabled), removing or not the temporary files (yes), using the result as new default focal length and pixel sizes (yes).

Index files

Astrometry.net needs index files to run. We strongly recommend you use the latest index files available from their website, i.e. the 4100 and 5200 series. The field of view of each series is described in their github page. (the official documentation does not yet include this table).

On Unix-based system, you can just follow along the instructions in the documentation.

On Windows, if you are running ansvr, those recent index files will not be made available by the Index Downloader. You can still download them separately and store them where the other index files are kept (would recommend to remove the old files, although it may mess up the Index Downloader).
How it works

Just like the internal solver, Siril will proceed with extracting the stars from your images (so as to benefit from internal parallelism) and submit this list of stars to astrometry.net solve-field. If you then want astrometry.net to crawl the index in parallel, you will need to specify it through the astrometry.cfg file.

11.1.4 Star detection

By default, the star detection uses the findstar algorithm with the current settings. It works very well to find many stars, but in some occasions we would like to detect the stars manually, or simply view which are used. A first step would be to open the PSF window and launch star detection, then adjust the settings (see the related documentation documentation).

Another approach would be to select the stars one by one by surrounding them with a selection then via a right click, choose Pick a Star. The more stars selected, the more likely the algorithm is to succeed.

Then in the astrometry window, expand the star detection section and activate the Manual detection. Instead of running findstar, it will use the current list of stars.

11.1.5 Understanding the results

When an astrometric solution is found, we can see in the Console tab this kind of messages:

<table>
<thead>
<tr>
<th>232 pair matches.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inliers: 0.996</td>
</tr>
<tr>
<td>Resolution: 0.196 arcsec/px</td>
</tr>
<tr>
<td>Rotation: -115.21 deg (flipped)</td>
</tr>
<tr>
<td>Focal length: 3959.95 mm</td>
</tr>
<tr>
<td>Pixel size: 3.76 μm</td>
</tr>
<tr>
<td>Field of view: 31' 15.46&quot; x 20' 51.09&quot;</td>
</tr>
<tr>
<td>Saved focal length 3959.95 and pixel size 3.76 as default values</td>
</tr>
<tr>
<td>Image center: alpha: 21h32m41s, delta: +57°36'22&quot;</td>
</tr>
<tr>
<td>Flipping image and updating astrometry data.</td>
</tr>
</tbody>
</table>

The astrometric solution gives us the J2000 equatorial coordinates of the image center, the projected horizontal and vertical dimension of the image on the sky, the focal length that could give this field for the given pixel size and consequently the actual image sampling, the angle the image makes with the north axis and some information about how many stars could be used to achieve the solution.

If it fails, check that start coordinates and pixel size are correct and try changing the input focal length from a factor 2, this will change the amount of stars downloaded from the catalogs, and maybe more stars will be identified. If Siril’s plate solve won’t find a solution, it is still possible to use an external tool to do it, the solution will be written in the FITS header either way.

11.1. Platesolving
11.2 Аннотации

Annotations are glyphs displayed on top of images to depict the presence of known sky objects, like galaxies, bright stars and so on. They come from catalogues but can only be displayed on images for which we know which part of the sky they represent, images that have been plate solved and contain the world coordinate system (WCS) information in their header, so only FITS or Astro-TIFF files.

Plate solving, can be done within Siril in the Image Information → Image Plate Solver... entry, or using external tools like astrometry.net or ASTAP.

When a plate solved image is loaded in Siril, you can see the sky coordinates for the pixel under the mouse pointer displayed at the bottom right corner and the buttons related to annotations become available. The first button toggles on or off object annotations, the second the celestial grid and the compass.

11.2.1 Виды каталогов

Siril comes with a predefined list of catalogues for annotations:

- Каталог Мессье (M)
- Новый общий каталог (NGC)
- Index Catalogue (IC)
- Lynds Catalogue of Dark Nebulae (LdN)
• Sharpless Catalogue (Sh2)
• Star Catalogue (3661 of the brightest stars)

В дополнение могут быть использованы 2 пользовательских каталога:
• Пользовательский каталог объектов глубокого космоса
• Пользовательский каталог объектов Солнечной системы

11.2.2 Управление каталогами

Both these catalogues can be enabled/disabled for display in the Preferences menu → Astrometry tab.

A slider on the right side, allows you to easily navigate across the catalogue list.

The two user defined catalogues can also be purged (ie deleted) via the appropriate buttons.
The user catalogues (DSO, SSO or extra catalogues) are stored in the user settings directory and can be easily modified.

Their location depends on the operating system:

- for Unix-based OS they will be in ~/.config/siril/catalogue
- on Windows they are in %LOCALAPPDATA%\siril\catalogue.

The position of the compass on the image can be adjusted from the preferences too.

These annotation catalogues are for display purposes only. They are not used in astrometry or photometry tools, contrary to the star catalogues like NOMAD, which can now be installed locally too.

![Compass setup](image)

Рис. 7: Local Catalogue (NOMAD) setup

### 11.2.3 Поиск нового объекта

When the name of an object in the image is known (if not, see the Inverse Search section), it is possible to add it to annotations:

- with the image loaded and plate solved, type Ctrl + Shift + / or Search in the pop-up menu (right click).

A small search dialog will appear. In it, object names can be entered, then pressing Enter will send an online request to SIMBAD (for a star of Deep Sky Object) to get the coordinates of an object with such a name. If found, and not already in any catalogue, the object will be added to the Deep Sky user Catalogue.

The items of this catalogue are displayed in ORANGE while the objects from the predefined catalogues are displayed in GREEN.

From Siril version 1.2, we can now search for solar system objects too, using the Miriade ephemcc service. This is done by prefixing the name of the object to be searched by some keyword representing the type of object: a: for asteroids, c: for comets, p: for planets. Since they are moving objects, they can be added several times, and the request is done for the date of observation of the currently loaded image. The date is associated to the name in the user Solar System Catalogue. The items of this catalogue are displayed in YELLOW.
Рис. 8: Deep sky objects from user and predefined catalogues
Examples of valid inputs (not case sensitive):

- HD 86574 or HD86574 are both valid for this star
- c:67p or c:C/2017 T2 are valid forms for comets
- a:1 and a:ceres are both valid for (1) Ceres
- a:2000 BY4 is valid for 103516 2000 BY4
- p:4 or p:mars for Mars

11.2.4 Filling a Solar System user Catalogue: which SSO is in this field?

To answer the question *Is there any solar system object in my image?*, a special function does a request to an online server of the IMCCE too (SkyBoT) and displays the results in the console and in the image.

- with the image loaded and plate solved, right click/Solar System Objects, or in the command line you can use the solsys function.

It displays in RED all the Solar System objects in the field of view (if any are known and found of course). Objects magnitudes and equatorial coordinates for the image date are printed in the console.

These red annotations will be erased as soon as the Show Objects names button is toggled.

However, you may want to save any particular item in the User Solar System Objects Catalogue. It can be done by using the Search command for a solar system object as previously described.

This way, the saved item is displayed in YELLOW and will be displayed in any image that has this field of view by enabling the annotations.
Рис. 10: View with predefined/DSO/SSO
Примечание: Newly discovered objects, or some fast moving objects, will have their position misaligned with the image. This is often the case for comets for example, which can be an arcminute off. This happens because the orbital parameters of the object are not very well known or that they have not been updated recently in the system. If you are looking for an alternate computation of the coordinates of the known objects of the field, you might query manually the JPL Small Body identification tool.

11.2.5 The inverse search: what is this object?

Especially useful for photometry works, it is possible to identify a star or other objects in the image by drawing a selection around them, right clicking to bring up the context menu, and selecting the PSF entry. This will open the PSF window, and if it’s a star it will display the Gaussian fit parameters, but it will also display a Web link at the bottom left of the window: opening it will bring you to the SIMBAD page for the coordinates of the object and in many cases will give you the name of the object. SIMBAD doesn’t have all known objects, but the coordinates from the page can still be used as a starting point to look for the object in other online catalogues, for example Gaia DR3 (VizieR).

11.2.6 Дополнительные каталоги

Sometimes, users create their own catalogues, we can try to link them here to help everybody. They are user catalogues, so installing them requires either replacing the current user catalogue, or by manually merging their lines into a new file.

Список известных пользовательских каталогов:

• Variable stars, extracted from GCVS 5.1, discussed here in French, (file link).
This section introduces you to all the utilities related to photometry, first explaining the principles of photometry, then how it is used in Siril.

Siril is able to determine the magnitude of stars as well as its uncertainty. From there it is possible to study the variability of certain stars, exoplanets, or occultations. A light curve is also built at the end of the process.

**Warning:** For an unrestricted use of photometry in Siril, we recommend to install the gnuplot software. Without it, Siril can't build or display light curves.

### 12.1 Principles

Photometry is the science of the measurement of light. It aims to measure the flux or intensity of light radiated by astronomical objects. In Siril, photometry can be used to analyze the light curve of variable stars, transits of exoplanets or occultations of stars, or to calibrate colors in RGB images.

Aperture photometry is the method used. Its basic principle is to sum-up the observed flux in a given radius from the center of an object, then subtract the total contribution of the sky background in the same region (calculated in the ring between the inner and outer radii, excluding the deviant pixels), leaving only the flux of the object to calculate an instrumental magnitude. This is illustrated in the following figure.

The values of these settings can be changed in the Фотометрия section of preferences or using the setphot command. The aperture must contain all pixels of the object to measure, the annulus should by opposition not contain any of its pixels. By default, the aperture is adjusted for a target using twice the PSF's FWHM, but the annulus size is fixed. These values should be adjusted for a given sampling, and checked with care.

**Note:** The following text is a truncated and modified copy of the excellent MuniPack software documentation, from David Motl and released under the GNU Free Documentation License, whose sources are available here.
Рис. 1: Пример фотометрии экзопланеты в Siril.

**Measuring magnitude of an object**

The sum $S$ of pixels in a small area $A$ around an object is a sum of the object's net intensity $I$ plus background intensity $B \cdot A$:

$$S = I + B \cdot A \quad (12.1)$$

The values of $S$ and $B$ are derived from the source frame, the area $A$ is determined as the area of circle of radius $r$, where $r$ is the size of the aperture in pixels. It is then easy to compute the net intensity $I$ of an object in ADU:

$$I = S - B \cdot A \quad (12.2)$$

Supposing that the net intensity $I$ is proportional to the observed flux $F$, we can derive the apparent magnitude $m$ of the object, utilizing the Pogson's law:

$$m = -2.5 \log_{10} \left( \frac{I}{I_0} \right) \quad (12.3)$$

**Оценка ошибки измерения**

Once we have derived the raw instrumental brightness of an object, we will try to estimate its standard error. First of all, we will recall a few general rules that apply to the standard error and its propagation. This is a general rule for error propagation through a function $f$ of uncertain value $X$:

$$\text{Var}(f(X)) = \left( \frac{df}{dx} \right)^2 \text{Var}(X) \quad (12.4)$$
Рис. 2: Circles of the aperture photometry
Using this general rule, we derive two laws of error propagation. In the first case, the uncertain value $X$ is multiplied by a constant $a$ and shifted by a constant offset $b$. This law can also be used in the case where only a multiplication or only an offset occurs.

$$\text{Var}(aX + b) = a^2 \text{Var}(X)$$

(12.5)

The second law defines the error of a logarithm of uncertain value $X$:

$$\text{Var}(\log(\pm bX)) = \frac{\text{Var}(X)}{X^2}$$

(12.6)

Please note, that the $\log$ function here is the natural logarithm, while the Pogson's formula (see above) incorporates the base-10 logarithm. The following equation helps us to deal with this difference:

$$\log_b(x) = \frac{\log_k(x)}{\log_k(b)}$$

(12.7)

Putting these two equations together we get:

$$\text{Var}(\log_{10}(\pm bX)) = \frac{\text{Var}(X)}{X^2 \log(10)^2}$$

(12.8)

If we have two uncorrelated uncertain variables $X$ and $Y$, the variance of their sum is the sum of their variances, this equation is known as Bienaymé formula.

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$$

(12.9)

From this formula, we can also derive the standard error of a sample mean. If we have $N$ observations of random variable $X$ with sample-based estimate of the standard error of the population $s$, then the standard error of a sample mean estimate of the population mean is

$$SE_X = \frac{s}{\sqrt{N}}$$

(12.10)

Armed with this knowledge, we can start thinking about the estimation of standard error of object brightness. We will consider the following three sources of uncertainty: (1) random noise inside the star aperture that includes the thermal noise of the detector, read-out noise of the signal amplifier and the analog-to-digital converter, (2) Poisson statistics of counting of discreet events (photons incident on a detector) that occur during a fixed period of time and (3) the error of estimation of mean sky level.

For the estimation of mean sky level, we have used the robust mean algorithm. It allows to estimate its sample variance $\sigma_{pxl}^2$. This is a pixel-based variance and because we have summed together $A$ pixels in the star aperture, the Bienaymé formula applies, the sum $S$ is a sum of $A$ uncorrelated random variables, each of which has variance $\sigma_{pxl}^2$. For the variance of the first source of error we get:

$$\sigma_1^2 = A \sigma_{pxl}^2$$

(12.11)

where $A$ is a number of pixels in the star aperture.

From Poisson statistics we can derive a variance that occur due to counting of discreet events, photons incident on a detector, that occur during a fixed period of time, the exposure. We will again need to use the gain $p$ of the detector to convert a intensity in ADU to a number of photons. If the measured net intensity of an object is $I$ we compute the mean number of photons $\lambda$ as

$$\lambda = I \ p$$

(12.12)
Then, the variance of intensity due to Poisson statistics is equal to its mean value.

\[ \sigma_{ph}^2 = \text{Var}(\text{Pois}(\lambda)) = \lambda = I_p \]  

(12.13)

The variance is in photons, we have to convert it back to ADU to get the variance in units \( ADU^2 \).

\[ \sigma^2 = \frac{\sigma_{ph}^2}{p^2} = \frac{I_p}{p^2} = \frac{I}{p} \]  

(12.14)

We have derived the sky level as a sample mean of pixel population in the sky annulus. Because each pixel in the annulus has variance \( \sigma_{pixel}^2 \), the variance of sample mean is

\[ s_{sky}^2 = \frac{\sigma_{pixel}^2}{n_{sky}} \]  

(12.15)

where \( n_{sky} \) is the number of pixels in sky annulus.

From equation (12.9) we compute the variance of object's intensity as

\[ \sigma_{ADU}^2 = \sigma_1^2 + \sigma_2^2 + A^2 s_{sky}^2 \]  

(12.16)

Note, that in equation (12.2) the sky level is multiplied by \( A \), so we have to multiply its variance by \( A^2 \) - see the equation (12.16). Now, we use the law of error propagation for the logarithm adopted to match the formula of the Pogson's law.

\[ \sigma_{mag}^2 = \left( \frac{-2.5}{\log(10)} \right)^2 \sigma_{ADU}^2 \]  

(12.17)

Putting equations (12.17) and (12.16) together, we can derive the standard error of the object's brightness in magnitudes as

\[ \sigma_{mag} = \frac{1.08574}{I} \sqrt{\sigma_{ADU}^2} \]  

(12.18)

12.2 Быстрая фотометрия

12.2.1 Photometry on hand-picked objects of a single image

The quick photometry button is a button located in the toolbar and used to perform a photometry of the stars, this is generally the simplest way to proceed.

Совет: If the star is in the middle of several stars and the tool fails to point to the right star, an alternative solution is to draw a selection around the star and then right-click and click on PSF. It may also be interesting to know that the middle click draws a selection of a recommended size for PSF/photometry (based on the configured outer radius).

Совет: When photometry is performed on the RGB layer, the results are actually calculated on the green layer. To obtain photometry on the red or blue layers, you need to work on the corresponding channels.
Performs a PSF (Point Spread Function) on the selected star and display the results. If provided, the \texttt{channel} argument selects the image channel on which the star will be analyzed. It can be omitted for monochrome images or when run from the GUI with one of the channels active in the view.

Click on this button to change the image selection mode, then click on a star. The photometry and the PSF (Point Spread Function) of the star are computed, giving plenty of details.

Two models are used for the calculation of the PSF, which can be selected by the user in the \textit{Dynamic-PSF} window (Ctrl + F6).

Рис. 3: Photometry results window.

The result of the photometry and the associated PSF are displayed in the form:

\begin{verbatim}
PSF fit Result (Gaussian, monochrome channel):

Centroid Coordinates:
\begin{align*}
x_0 &= 5258.25 \text{px} & 09h25m34s & J2000 \\
y_0 &= 2179.72 \text{px} & +69^\circ 49'31'' & J2000 \\
\end{align*}

Full Width Half Maximum:
\begin{align*}
\text{FWHM}_x &= 7.13'' \\
\text{FWHM}_y &= 6.79''
\end{align*}
\end{verbatim}
1. The fit was done with the Gaussian fitting function so no additional parameters are needed. However, if Moffat was used, the following output will be shown:

<table>
<thead>
<tr>
<th>PSF fit Result (Moffat, beta=2.9, monochrome channel):</th>
</tr>
</thead>
</table>

2. **Centroid Coordinates** gives the coordinates of the centroid in pixels. However, like in the example above, if astrometry was set on the image, Siril gives coordinates in the World Coordinate Systems (RA and Dec).

3. **Full Width Half Maximum** (FWHM) is returned in arcsec if the image scale is known (obtained from its header or from the GUI: menuselection: Image information --> Information) and in pixels if not. The roundness $r$ is also computed as the ratio of \( FWHM_y / FWHM_x \).

4. **Angle** is the rotation angle of the $X$ axis with respect to the centroid coordinates. It varies in the range $[-90, +90]$.

5. **Background Value** is the local background in the $[0, 1]$ range for 32-bits images and $[0, 65535]$ for 16-bits images. This is a fitted value, not the background computed in the aperture photometry annulus.

6. The **maximum Intensity** value is also a fitted value and represents the amplitude. It is the maximum value of the fitted function, located at the centroid coordinates.

7. The **magnitude**, given with its uncertainty, is the result of photometry. However, if for some reasons the calculation cannot be done (saturated pixels or black pixels), an uncertainty of $9.999$ is given. In this case, the photometry is flagged as invalid but a magnitude value is still given, although it should be used with caution.

8. An estimator of the **signal-to-noise ratio** is shown in the results. Its value is calculated from the following formula and given in dB:

$$ SNR = 10 \log_{10} \left( \frac{I}{N} \right) $$

(12.19)

where $I$ is the net intensity, proportional to the observed flux $F$ and $N$ the total of uncertainties as expressed in (12.18).

For easier understanding, it is associated with 6 levels of quality:

1. Отлично (сигнал/шум > 40 дБ)
2. Хорошо (сигнал/шум > 25 дБ)
3. Удовлетворительно (сигнал/шум > 15 дБ)
4. Мало (сигнал/шум > 10 дБ)
5. Плохо (сигнал/шум > 0 дБ)
6. Н/Д

Последнее обозначение отображается, только если вычисление не удалось по той или иной причине.

9. Finally, RMSE gives an estimator of the fit quality. The lower the value, the better the result.

When the image is plate-solved, the button More details at the bottom of the window links to a page on the SIMBAD website with information about the selected star. However, it is possible that the page does not give any additional information if the star is not in the SIMBAD database.

Рис. 4: More details about the analyzed star. Click on the picture to enlarge.

12.2.2 Quick photometry on sequences

Quick photometry can also be performed on a sequence. This is generally intended to obtain a light curve as explained here. To proceed, you must load a sequence, make a selection around a star, then right click on the image.

Совет: Ideally, the sequence must be registered without interpolation so as not to alter the raw data. For example, use the global registration with the option Save transformation in seq file only.

Примечание: Make sure the inner and outer radii for the background annulus are adapted to the star and sequence being analyzed. Some images may have much larger FWHM than the reference image, because of sky conditions or bad tracking. They can be changed in the preferences or with the setphot command.
At the end of the process, Siril automatically opens the plot tab showing computed curves. It is possible to click on several stars to reproduce the calculation, however the first star keeps the particular status of *variable*, and the others serve as *references*. This is important in the calculation of the light curve.

![Image of stars showing variable and references](image)

Рис. 5: In this example, 3 stars have been analyzed. The first one is used as variable. The others are references.

### 12.2.3 Computing true magnitudes

The calculated magnitude is only meaningful if it is compared to others in the linear image. Indeed, the value given does not correspond at all to the true visible magnitude of the star, it is uncalibrated, also called relative magnitude.

Siril provides tools that can be used to calculate an approximate apparent magnitude. This requires knowing the magnitude of another star visible on the image that will act as reference. It is currently possible to use only a single star as reference, hence the *approximate*. For a greater precision, use a star of similar color and magnitude as the star(s) you want to measure should be chosen, and its provided magnitude should be in adequation with the filter used to capture the image. Catalogs contain magnitudes computed using a photometric filters, which is generally not what amateur use to make nice pictures, this adds another approximation.

- Do a quick photometry on a known star, the given relative magnitude is -2.428. It is possible to find out the actual visible magnitude by clicking on the *More details* button as explained above. Let’s say the value found is 11.68 (make sure you use a value corresponding to the spectral band of the image).

- Once done, keep the star selected, then enter the following command in Siril

```
setmag 11.68
```

That will output something like

Командная строка Siril

setmag magnitude

Calibrates the magnitude by selecting a star and giving the known apparent magnitude.

All PSF computations will return the calibrated apparent magnitude afterwards, instead of an apparent magnitude relative to ADU values.
To reset the magnitude constant see UNSETMAG

Links: psf, unsetmag

- Now, all calculated magnitudes must have values close to their true visual magnitude. However, this is especially true for stars whose magnitude is of the same order of magnitude as the star taken as reference.
- To unset the computed offset, just type

  unsetmag

Командная строка Siril

unsetmag

Сбрасывает калибровку блеска к 0. См. SETMAG

Links: setmag

Совет: The same commands exist for the sequences. They are seqsetmag and sequnsetmag. It is used in the same way when a sequence is loaded.
Приложение 6: Окно результатов фотометрии с установленным истинным значением magnitude.
This command is only valid after having run SEQPSF or its graphical counterpart (select the area around a star and launch the PSF analysis for the sequence, it will appear in the graphs). This command has the same goal as SETMAG but recomputes the reference magnitude for each image of the sequence where the reference star has been found. When running the command, the last star that has been analyzed will be considered as the reference star. Displaying the magnitude plot before typing the command makes it easy to understand. To reset the reference star and magnitude offset, see SEQUNSETMAG.

Links: setmag, seqpsf, psf, sequnsetmag

#### 12.3 Световые кривые

In astronomy, a light curve is a graph of light intensity of a celestial object as a function of time, typically with the magnitude of light received on the y axis and with time on the x axis. Siril is able to generate such curves when analyzing stars.

Выбор переменной и опорной (звезды сравнения) звёзд возможен двумя способами: вручную или используя список звёзд, полученных с помощью плагина экзопланет N.I.N.A.

### 12.3.1 Выбор звезды вручную

Start by selecting stars and running photometry analysis on the sequence for each, as explained [here](#).

**Предупреждение:** Make sure to not select variable stars for references. If the astrometry is done on your image, do not hesitate to use the SIMBAD request to know more about the stars.

**Совет:** It is preferable to choose references whose magnitude is close to that of the variable.

Once done, Siril automatically loads the Plot tab as shown in the figure below. This shows FWHM curves expressed as a function of frame number.

What interests us in this part is to display the magnitude curves. Simply go to the drop-down menu and change FWHM to Magnitude. The magnitude curves of each analyzed star are then displayed. This also
Рис. 7: One star is the variable (purple with a V) and the 5 others are used as references.
Рис. 8: The plot tab as showed right after the quick photometry on sequence.
results in the button *Light Curve* being sensitive. It is also recommended to check the *Julian Date* button in order to plot magnitude as a function of a date.

Once the analysis is completed with a number of reference stars of at least 4 or 5 (the higher the number, the more accurate the result), you can click on the *Light Curve* button. Siril will ask for a file name to save the data in *csv* format, then the light curve will be displayed in a new window. The *csv* file can of course be used in any other software or website to reduce the data.

**Предупреждение**: As already mentioned, the software gnuplot must be installed to be able to see light curves.
12.3.2 NINA exoplanet button

In order to automate the process of transit analysis of exoplanets, lists of reference stars, also called comparison stars, could be obtained from star catalogues, with the appropriate criteria: similar magnitude, similar color (to not change their relative magnitude with atmospheric extinction at different elevations), proximity.

The capture software N.I.N.A has an exoplanet plugin that will show such stars and allow the list to be saved in a CSV file, such as csv file:

| 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, 22.70972222222222 | | | | |
| Var, HQ Vul, 3.8351043206620834, 157, 1690, 49.393939393939391, 2104, 1905.745454545454545454, 300.55808, 22.64067 | | | | | |
| Comp1, ATO J300.3222+22.7056, 2.4268101078425852, 1367, 1465, 352, 4496, 1913.9504132231405, 300.3229415181337, 22.70568145378887 | Comp1, HD 189657, 2.5343988482845927, 2527, 2808, 23.814814814814813, 2096, 1908.5061728395062, 300.08714683055996, 22.4400393728 | | | | |
| Comp2, 000-BJP-946, 2.2738807043120195, 1832, 750, 29.962962962962962962, 2024, 1910.0648148148148, 300.237416666666666666, 22.846999999999998 | Comp2, 000-BJP-942, 2.0977710589704297, 2760, 1572, 31.083333333333332, 2096, 1908. | | | | |

(continues on next page)
In the Plot tab, Siril can load this file using the NINA exoplanet button. To use this, a few prerequisites must be met:

- the sequence of calibrated images must be already loaded
- the reference image of the sequence must be plate solved, to make sure we identify the correct stars from their equatorial J2000 coordinates
- gnuplot is installed to create or show the light curve, otherwise only the data file will be created.

From there, everything is automatic, showing the light curve for the selected comparison stars at the end of the process.

The following video shows an automated processing of light curve with comparison star list from NINA:

### 12.3.3 Commands and automatic operation

It is also possible to automate or create the light curve remotely using the `light_curve` command. As blind operation needs as much automation as possible, the configuration of the background annulus radii can be automated with the `-autoring` argument: it runs a star detection in the reference image and multiplies the mean FWHM with a configurable factor to obtain the inner and outer radii that should work with the sequence.

Командная строка Siril

```bash
light_curve sequencename channel [-autoring] { -at=x,y | -wcs=ra,dec } { -refat=x,y | -refwcs=ra,dec } ...  
light_curve sequencename channel [-autoring] -ninastars=file
```

Analyse several stars with aperture photometry in a sequence of images and produce a light curve for one, calibrated by the others. The first coordinates (in degrees) are for the star whose light will be plotted, the others for the reference stars. Alternatively, a list of target and reference stars can be passed in the format of the NINA exoplanet plugin star list, with the `-ninastars=` option. Siril will verify that all reference stars can be used before actually using them. A data file is created in the current directory named `light_curve.dat`, gnuplot plots the result to a PNG image if available.

The ring radii for aperture photometry can either be configured in the settings or set to a factor of the reference image's FWHM if `-autoring` is passed.
Siril has several tools that can help you to analyze your image and tell you about the quality of the shot. In particular if your setup has or not optical defects.

### 13.1 Наклон

The first tool proposed by Siril is the tilt calculation. Sensor tilt occurs when the sensor is not orthogonal to the imaging plane: this requires an intervention on the optical system. You can execute this functionality in two different ways. Either via the GUI (clicking on the tilt button on the Dynamic PSF Window, Ctrl + F6) or via the command line. The latter even offers an alternative that allows you to calculate the tilt over a whole sequence of images for greater accuracy. The following command:

<table>
<thead>
<tr>
<th>Командная строка Siril</th>
</tr>
</thead>
<tbody>
<tr>
<td>tilt [clear]</td>
</tr>
</tbody>
</table>

Computes the sensor tilt as the fwhm difference between the best and worst corner truncated mean values. The clear option allows to clear the drawing

will output:

```
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 the console are indicated:
• the number of stars used for the measurement
• the average FWHM on the sensor, free of outliers
• the tilt, expressed as the difference between the best and the worst FWHM on the 4 corners of the image with in parenthesis the percentage of tilt deviation (value greater than 10% indicates a tilt problem)
• the aberration, expressed by the difference in FWHM between the stars in the center and the stars on the edges of the sensor

If the number of stars detected is low (<200), the dynamic PSF detection parameters allow improvement by adjusting the threshold / radius. In fact, the greater the number of stars used in the calculation, the more reliable the result of the analysis.

Предупреждение: For the result to make sense, it is preferable to run this command on a sub and not a stacking result. A pre-processed image (just demosaiced for color sensors) is therefore ideal. Moreover, the drawn quadrilateral has its proportions exaggerated, in order to be more visible on the screen. It can’t correspond exactly to reality.

Рис. 1: Display of the tilt diagram

Совет: As well as using the `tilt -clear` command, the tilt diagram can be cleared using the `Remove` button in the Dynamic PSF dialog.
Командная строка Siril

```
seqtilt sequencename
```

Same command as TILT but for the loaded sequence or the sequence `sequencename`.

It generally gives better result

Links: `tilt`

### 13.2 Аберрации

This tool creates a 3x3 mosaic of the image center, corners and edges. This makes it easy to compare the shape of the star in different sections of the image. You can access this feature by right-clicking on the image and selecting *Aberration Inspector*. You can change the settings of this tool, to modify the size of the panels and the window, in the *preferences*.

![Aberration Inspector](image)

Рис. 2: Aberration inspector window showing aberration in the stars located in the edges because of the optical system.
It is also a very good indicator to know if the image contains a gradient: the differences in brightness becoming very visible.

Рис. 3: Aberration inspector window showing differences in brightness.

Командная строка Siril

```
inspector
```

Создаёт из текущего изображения мозаику из 9 изображений, отображающую углы и центр изображения для более тщательного изучения
This is a documentation for Siril's statistics, given by the graphical user interface (GUI) from the contextual menu of images (right-clicking in it) then selecting Statistics..., from the Image Information submenu of the application menu after also selecting Statistics... or using the stat command. Note that when using the GUI, it is possible to draw a selection in the loaded image and that when doing so, the statistics are computed on the pixels of region.

The option Per CFA channel allows you to calculate statistics for each R, G and B channel in CFA images, even if the image has not been demosaiced.

Many of these values are measures of statistical dispersion.

**Командная строка Siril**

```bash
stat [-cfa] [main]
```

Returns statistics of the current image, the basic list by default or the main list if main is passed. If a selection is made, statistics are computed within the selection. If -cfa is passed and the image is CFA, statistics are made on per-filter extractions.
Рис. 1: One channel statistics for CFA image. The given values are not really relevant in this case.

Рис. 2: Three channel statistics for CFA image.
14.1 Estimators

14.1.1 Среднее

This is the arithmetic mean, also known as average or arithmetic average. This is computed by doing the sum of the pixel values divided by the number of pixels in an image channel.

14.1.2 Медиана

The median is the value separating the higher half from the lower half of a dataset. Generally, it represents the value of the background of an astronomical image.

14.1.3 Сигма (\(\sigma\))

Also known as the standard deviation, noted \(\sigma\), this is a measure of dispersion of the image pixels based on squared differences from the average. The sigma value of a sub image containing only the background will represent the noise of the image.

14.1.4 Background noise

This estimator is available by the GUI from the Image Information submenu of the application menu after selecting Noise estimation, and is also displayed at the end of stacking.

This is a measure of estimated noise in image background level, for pixels having a value low enough to be considered as background. It is an iterative process based on \(k\sigma\) (a factor of the standard deviation above the median), so there is no fixed threshold for the low enough.

Командная строка Siril

bgnoise

Возвращает значение уровня фонового шума изображения, загруженного в память

14.1.5 Абсол. откл.

The Average Deviation, also called AAD for average absolute deviation or mean absolute deviation. In order to understand what the average deviation is, one needs to understand what the term absolute deviation is. Absolute deviation is the distance between each value in the dataset and that dataset's mean (in this instance) or median (for MAD below). Taking all of these absolute deviations, finding the average, and the mean average deviation is computed. To simplify, if standard deviation is the squared deviation from the mean, this is the linear version of it.
14.1.6 MAD

The Median Absolute Deviation is a robust measure of how spread out a set of data is. The absolute deviation and standard deviation are also measures of dispersion, but they are more affected by extremely high or extremely low values. It is similar to the average deviation above, but relative to the median instead of the mean.

14.1.7 BWMV

The biweight midvariance is yet another tool to measure dispersion of a dataset, even more robust than others cited above to outliers. It discards the data points too far way from the median and computes a weighted variance, weights decreasing as the data points are further way from the median. The estimator of dispersion is the square root (marked as $\sqrt{BW MV}$) of this value.

14.1.8 Location and Scale

These parameters, often colloquially called scale and offset, are not displayed in the user interfaces but are computed internally by Siril. In order to align the histograms of the different images for normalization before stacking, one needs to compute where they are in terms of level and how wide they are in terms of spread. A valid estimator of location could be taken as the median while the MAD or the $\sqrt{BW MV}$ could be used for scale. However, in order to give more robustness to the measures, the pixels more than $6 \times$ MAD away from the median are discarded. On this clipped dataset, the median and $\sqrt{BW MV}$ are re-computed and used as location and scale estimators respectively. They are computed relative to the reference image of a sequence in Siril.
Siril has a command line in its graphical user interface and an ability to run scripts that are a list of commands, either from the graphical user interface or from the command line interface. In general, commands that modify a single image work on the currently loaded image, so the use of the `load` command is required in scripts, and commands that work on a sequence of images take the name of the sequence as argument. If files are not named in a way that Siril detects as a sequence, the command `convert` will help.

**Совет:** The `Space` character is the delimiter between arguments. If you need to have spaces inside the arguments, you can use the quote or double quote, just like in a shell.

Commands can be typed in the command line at the bottom of Siril's main window. Another way is to put commands in a file and execute it as a script. To execute the file from the GUI, add it to the configured script directories or from the GUI, use the `@` token of the command line like so:

```
@file_name
```

Some commands (preprocess, stack, and all save commands) can use file names containing variables coming from the FITS header. The format of the expression is explained in details [here](#) and can be tested using the `parse` command.

### 15.1 Использование сценариев

Возможны три способа запуска сценария:

- from the graphical user interface, using the `@` keyword on the command line, followed by the script name in the current working directory,
- из графического интерфейса пользователя используя меню Сценарии,
- from the command line interface (`siril-cli` executable), using argument `-s` followed by the script's path (see the man page for more info).
The scripts menu only appears if some scripts have been found in the script search directories defined either by default or by the user in the preference menu.

### 15.2 Populating the list of scripts

By default, when Siril is installed, a number of scripts are automatically installed. These built-in scripts, the official ones, are developed by the development team and are guaranteed to work: they are meant to cover specific use cases.

#### 15.2.1 Adding custom scripts folders

You can, of course, write your own and tell Siril where to find them:

- Click on the Burger icon then on Preferences (or hit Ctrl+P).
- Click on the Scripts section.
- Copy to a new line the path to the location to store them (create a folder on your computer as required or point to an existing one).
- Click on the Refresh icon just below.
- Нажмите Применить.

You can have as many user-defined folders as you wish, just add them to the list.

If you have just added a new script in one of the folders and wish to refresh the menu, type the command *reloadscripts* in the command line or open the Preferences → Scripts section and use the Refresh icon. This scans all the folders of the list and find all the files with the *.ssf* extension.

**Warning:** It is strongly advised not to store your custom scripts within the same folder as Siril built-in scripts. On Windows, they may get wiped when installing a newer version or prevent correct uninstall. On MacOS, it will break the bundle and prevent using Siril altogether.

Don't worry, as the list of scripts locations is stored in your configuration file, you should find them back when installing a newer version.
15.2.2 Troubleshooting

For different reasons, it is possible that the Scripts menu is not visible. This means that the scripts have not been found. If this is the case, please use the following procedure.

- Click on the Burger icon then on Preferences.
- Click on the Scripts section.
- Delete all the lines in the field Script Storage Directories as shown in the illustration below.
- Нажмите Применить.
- Закрыть и перезапустить Siril.

Рис. 1: Script page of preferences. The script are loaded from the paths listed in the Script Storage Directories.
15.3 Встроенные сценарии

All built-in scripts must follow this file structure:

- **Mono_Preprocessing.ssf**: script for monochrome DSLR or Astro camera preprocessing, uses biases, flats and darks, registers and stacks the images. To use it: put your files (RAW or FITs) in the folders named *lights, darks, flats* and *biases* (in the Siril default working folder), then run the script.

- **OSC_Preprocessing.ssf**: same script as above but for One-Shot Color (OSC) DSLR or Astro camera. To use it: put your files (RAW or FITS) in the folders named *lights, darks, flats* and *biases* (in the Siril default working folder), then run the script.

- **OSC_Extract_Ha.ssf**: script for OSC DSLR or astro camera preprocessing, for use with Ha filter or dual-band filter. This script extracts the Ha layer of the color image. To use it: put your files (RAW or FITs) in the folders named *lights, darks, flats* and *biases* (in the Siril default working folder), then run the script.

- **OSC_Extract_HaOIII.ssf**: same script as above, but extracts Ha and OIII layers of the color image. To use it: put your files (RAW or FITs) in the folders named *lights, darks, flats* and *biases* (in the Siril default working folder), then run the script. You can also use the menu *Image Processing* then *RGB compositing* and put Ha result in Red channel and OIII result in Green and Blue layers to get an HOO image.

- **RGB_Composition.ssf**: This script added in version 1.2 registers monochrome images with a global registration, reframes them to their common area, and takes the first three images to create a color image. The input images should be put alone in a directory and named *R.fit* (or with the configured extension), *G.fit* and *B.fit*. The result will be named *rgb.fit*. Make sure you remove the *process* directory between each run.
Язык сценариев

At the beginning of the scripts, and thanks to the contribution of a user, the scripts existed in two versions (English, and French). When Siril 1.2.0 was released, it was decided to keep only the English scripts for simplicity of maintenance. We encourage users to distribute translations of the official scripts to their respective communities if they deem it necessary.

15.4 Написание собственных сценариев

A script file is a simple text file with the extension *.ssf.

Writing a script is not difficult. It is a succession of calls to commands that will be executed sequentially. Each command must be executed without returning an error, otherwise the script stops. It is therefore strongly recommended to use the list of commands to know the syntax and the number of parameters needed. Also, some commands are not scriptable and are indicated with the icon. It can also be useful to test each script line in the Siril command line.

Each new script created in this way should be placed in a user-defined folder for Siril to find them.
Siril can both operate with its graphical user interface (GUI) and with a command line interface (CLI) that does not even require having a display. It can process images for other programs, on remote or embedded computers, using either scripts or real-time generated operations called commands. The capabilities of the headless Siril are in fact those of the available commands. There are more than a hundred, allowing preprocessing, processing and photometry analysis to be done automatically.

Commands can also be used in the GUI version of Siril, either from the embedded command line at the bottom of the control panel, or with scripts. Scripts are simply a text file containing a list of commands. Reading the scripts page is recommended before going further.

With the headless version, commands can be executed either by passing a script to run, or by setting the standard input as a script and writing commands to it, with -s - command line option, or using some named pipes.

### 16.1 Command Stream (Pipe)

This mode has been introduced with Siril 0.9.10. Commands can be sent through a named pipe and logs and status can be obtained through another. The mode is activated using the -p command line argument.

The protocol is quite simple: Siril receives commands and outputs some updates. Only commands that are marked as scriptable can be used with this system. Whenever the command input pipe is closed or the cancel command is given, the running command is stopped as if the stop button was clicked on in the GUI. The pipes are named `siril_command.in` and `siril_command.out` and are available in `/tmp` on Unix-based systems. Since version 1.2.0, the paths of the pipes can be configured with -r and -w options, which allows external programs to create them before starting Siril, typically with the `mkfifo` command. Also new in 1.2.0, a `ping` command will simply give a status return, indicating if siril is ready to process a command or already busy.

Outputs of siril on the pipe is a stream of one line text and formatted as follows:

- **ready** is printed on startup, indicating siril is ready to process commands
- **log**: followed by a log message
• **status**: verb [subject], where verb can be either of starting, success, error or exit (exit message is not yet implemented). The subject is the current command name, except for exit that indicates that siril suffered a fatal error and has to exit.

• **progress**: value% is the equivalent of the progress bar, it sends percents periodically, and sometimes 0% at the end of a processing as an idle information.
17

Разбор путей

Parsing is the ability to parse, i.e. to write strings based on data contained in the FITS header. Path parsing, introduced in Siril 1.2.0, aims at giving more flexibility to scripting by using header data to write/read filenames or paths. For now, this is used with the following commands:

- `calibrate`
- `stack` and `stackall`
- all the `saveXXX` commands

and of course, their GUI counterparts.

17.1 Syntax example

We will take an easy example to begin with. Say you have a file named `light_00001.fit` and you would like to locate a masterdark from your masters library that matches the characteristics of said light. As you have chosen a convention to name your masterdarks, you know that the correct dark should be named something like:

```
DARK_"exposure"s_G"gain"_O"offset"_T"temperature"C_bin"binning".fit
```

with the terms between quotes replaced with the values read from your light header. For an exposure of 120s, temperature of -10C, gain/offset of 120/30 and binning 1, the masterdark would be named:

```
DARK_120s_G120_O30_T-10C_bin1.fit
```

Well, that's exactly what this feature allows to do. If you specify the dark name with the conventions explained just after, you can tell Siril to open the light image, read its header and use its values to write such string (and then use it to calibrate your light).

You can read the info contained in the header either through `dumpheader` command or by right-clicking on an opened image and selecting `FITS Header`. You would normally get a print like the one below (some keys removed for brevity):
The format used to specify the dark name would then be:

```
DARK_${EXPTIME:%d}s_G$GAIN:%d$_O$OFFSET:%d$_T$SET-TEMP:%d$C_bin$XBINNING:%d$.fit
```

All the terms to be parsed are formed as follows: $KEY:fmt$

- **KEY** is any (valid) key from the light FITS header
- **fmt** is a format specifier.

For instance, $EXPTIME:%d$ will be parsed to **120** if the light have been exposed for **120**s. But would be parsed to **120.0** if you specify $EXPTIME:%0.1f$, thanks to the formatter **%x.yf**.

The full expression above would therefore evaluate to:

```
DARK_120s_G120_O30_T-10C_bin1.fit
```

In this first example, we have only used conversion to integers with **%d**. But there are other conventional formatters that you can use:

- **%x.yf** for floats
- **%s** for strings

**Примечание**: For strings, leading and trailing spaces are always removed, while spaces within the strings
You can also use some less conventional formatters:

- In order to parse a date from a date-time header key, you can use the special non-standard formatter \texttt{dm12}, which means date minus 12h. In the header above, the key DATE-OBS has a value of '2022-01-24T01:03:34.729'. $\texttt{DATE-OBS:dm12}$ would convert to '2022-01-23', which was the date at the start of the night. You can also use special formatter \texttt{dm0} which will just parse the date, without substracting 12h.

- In order to parse a date-time from a date-time header key, you can use the special non-standard formatter \texttt{dt}, which just means date-time. In the header above, the key DATE-OBS has a value of '2022-01-24T01:03:34.729'. $\texttt{DATE-OBS:dt}$ would then convert to '2022-01-24_01-03-34'.

- In order to parse RA and DEC info from OBJCTRA and OBJCTDEC header keys, you can use the special non-standard formatters \texttt{ra} and \texttt{dec}. In the header above, the keys OBJCTRA and OBJCTDEC have a value of '06 30 36' and '+04 58 51' respectively. $\texttt{OBJCTRA:ra}$_$\texttt{OBJCTDEC:dec}$ would convert to '06h30m36s_+04d58m51s'.

- In order to parse RA and DEC info from RA and DEC header keys, when in decimal format, you can use the special non-standard formatters \texttt{ran} and \texttt{decn}. In the header above, the keys RA and DEC have a value of '97.6960081674312' and '4.99212765957446' respectively. $\texttt{RA:ran}$_$\texttt{DEC:decn}$ would convert to '06h30m47s_+04d59m32s'.

To test the syntax, you can load an image and use the \texttt{parse} command, as reminded below.

Командная строка Siril

\begin{verbatim}
parse str [-r]
\end{verbatim}

Parses the string \texttt{str} using the information contained in the header of the image currently loaded. Main purpose of this command is to debug path parsing of header keys which can be used in other commands. Option \texttt{-r} specifies the string is to be interpreted in read mode. In read mode, all wilcards defined in string \texttt{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.

If \texttt{str} starts with \texttt{$def} prefix, it will be recognized as a reserved keyword and looked for in the strings stored in gui\_prepro.dark\_lib, gui\_prepro.flat\_lib, gui\_prepro.bias\_lib or gui\_prepro.stack\_default for \texttt{$defdark}, \texttt{$defflat}, \texttt{$defbias} or \texttt{$defstack} respectively.

The keyword \texttt{$seqname$} can also be used when a sequence is loaded.
17.2 Finding a file with path parsing

In the example above, we have seen that we could find the name of a masterdark based on the information contained in the header of the light to be calibrated. This is what is called in the `parse` command, the `read` mode.

This behavior is mainly used in conjunction with the `calibrate` command/tab. In the `-dark=` option of the command or in the `Dark` field of the GUI, you can use the syntax described above. So that you are sure that the matching dark will be fetched to calibrate the light. The same is equally applicable to Bias and Flat. You can, of course, also give a full (or relative) path to the file. And the path can also contain expressions of the same kind.

For instance, for flats, you may want to specify the path to a library, which could contain filter or telescope information, as you may have multiple setups. A path like:

```
~/astro/masters/flats/$INSTRUME:%s$_$TELESCOP:%s$/FILTER:%s$/FLAT_bin$XBINNING:%d$.fit
```

Would then evaluate to:

```
~/astro/masters/flats/ZWO_ASI294MC_Pro_61EDPH/DualBand/FLAT_bin1.fit
```

and is a valid value for Flat input.

Of course, if you were to write this as a command in your scripts or in the GUI Flat field every time you calibrate, that could become a bit tedious. That’s when reserved keywords come to the rescue. There are 3 reserved keywords for masters:

- `$defdark`
- `$defflat`
- `$defbias`

Their values are set in `Preferences → Pre-processing section`. You can also specify them through scripting thanks to `set` command. They correspond to the values of `gui_prepro.dark_lib`, `gui_prepro.flat_lib` and `gui_prepro.bias_lib`.

When their values are set and you chose to use them as defaults, they will be displayed in the fields of the Calibration tab. You can also start to write your scripts using these keywords. The calibration step of a new generic script for a color camera could look like:

```
calibrate light -dark=$defdark -cc=dark -flat=$defflat -cfa -equalizecfa -debayer
```

This would pick your masters directly from your libraries and make sure you never mix them up during the calibration step.

17.3 Writing a file with path parsing

Now, while it is handy to be able to find the files, it would be equally useful to use this syntax to store your files while stacking. That is exactly what the `write` mode is about. The syntax can be used in the `-out=` field of `stack` and `stackall` commands, or in the corresponding field in the GUI.

Let’s say you want to write a generic script that prepares your master darks each time you renew your library. In the `stack` line of the script, you could write:

```
stack dark rej 3 3 -nonorm -out=$defdark
```
This line ensures that the resulting masterdark will be stored at the right location with the correct filename that can then be fetched to calibrate your lights.

In order to introduce even more flexibility with the stack commands, there are two more reserved keywords:

- \$defstack
- \$sequencename$

As for default masters, \$defstack is configured in the same section of the Preferences, or with a set command on gui.prepro.stack_default. For instance, let’s assume you have defined \$defstack as:

Result$_OBJECT:%s$_$DATE-OBS:dm12$_$LIVETIME:%d$s

The script line:

```
stack r_pp_light rej 3 3 -norm=addscale -output_norm -out=$defstack
```

would save the stack result as:

Result_Rosette_Nebula_2022-01-24_12000s.fit

As of Siril 1.2.0, the default name for stacking is defined as \$sequencename$stacked (the ‘_’ sign is added if not present). This is similar to the behavior in previous versions, except it is now explicit that pathparsing is used.

### 17.4 Using wildcards

It could be that you want to use some key value in your masters name that do not match the key value in the frames to be calibrated. With an example, it may be a bit clearer. Say, you want, in your masterflats names, to keep record of their exposure time. Something like:

FLAT_1.32s_Halpha_G120_O30_bin1.fit

If you put a field \$EXPTIME:%0.2f$ in \$defflat, it will end up with an error at calibration step. Simply because the EXPTIME key will be read from the light frame to be calibrated, not a flat.

To deal with this situation, you can use wildcards in the expressions to be parsed:

```
FLAT_*EXPTIME:%0.2f$_FILTER:%s$_GAIN:%d$_OFFSET:%d$_bin$XBINNING:%d$
```

Note the * symbol placed right before \textit{EXPTIME}.

What this symbol means is the following:

- \textbf{In write mode}, so basically when stacking your masterflat, the field \textit{EXPOSURE} will be used to form the filename to be saved. In the example above, you would then effectively save to FLAT_1.32s_Halpha_G120_O30_bin1.fit.

- \textbf{In read mode}, so when calibrating your lights, the field \textit{EXPOSURE} will be replaced by *.

  When searching for such file, Siril will fetch all the files that matches the pattern FLAT_*_Halpha_G120_O30_bin1.fit. Hopefully, your naming convention is robust enough so that it finds only one matching file and uses it to calibrate.
Предупреждение: В случае, если Siril найдет более одного файла в режиме чтения, оно исполнит предупреждение на консоль и выберет самый свежий файл. Поскольку это может не дать необходимого результата, вам следует рассмотреть вопрос о переименовании имен файлов.
Глава 18

Укладка на лету

Укладка на лету это техника астрофотографии, которая позволяет в реальном времени сложить серию изображений для получения изображения более высокого качества. В отличие от традиционного сложения изображений, которое включает объединение многих изображений после того, как они были получены, при укладке на лету объединяются по мере их получения. Благодаря этому возможен мгновенный предпросмотр окончательного изображения, что позволяет астрофотографу внести поправки в съёмочный процесс для улучшения окончательного результата.

Siril 1.2.0 имеет эту возможность, которая, на данный момент, является экспериментальной. Укладка на лету позволяет отслеживать файл в реальном времени и складывать изображения по мере их появления. Укладка возможна как с использованием калибровочных кадров (темновых кадров, кадров шума считывания и кадров плоского поля), так и без них. Калибровочные кадры должны быть сделаны заранее, если планируется их использовать.

18.1 Укладка на лету (интерфейс пользователя)

Примечание: Укладка на лету возможна только с использованием файлов в формате FITS или "сырых" (RAW) изображений с камеры.

Для начала укладки на лету необходимо:

- Указать рабочий каталог при помощи кнопки Домой, в котором будут сохраняться изображения по мере съёмки.
- Нажать кнопку, обведённую рамкой на изображении ниже.

Появится новое окно.
В этом окне есть несколько кнопок и настроек. Кнопка Пуск, которая после нажатия становится кнопкой Пауза, и кнопка Стоп. Первая кнопка запускает или приостанавливает наблюдение за рабочим каталогом, а вторая кнопка — останавливает наблюдение.

Все остальные настройки стандартны для предобработки астрономических изображений:

- **дебайеризация**: Шаблон Байера определяется в файлах и дебайеризация включается автоматически. Это более индикатор, чем настройка.

- **использовать 32 бита**: Для обработки изображений будет использована 32 битная глубина цвета. Это медленнее и, в общем, в терминах качества, для укладки на лету бесполезно.

- **удалить градиент**: Применить удаление линейного градиента фона на откалиброванных входящих кадрах.

- **регистрация по сдвигу**: При регистрации изображения только сдвигаются, без использования вращения, что сильно ускоряет обработку изображений. Для альт-азимутальных монтировок или монтировок с большим дрейфом эту настройку необходимо отключить.

Ниже приведены 3 раздела, в которых отображается информация, необходимая для оценки процесса укладки.

- **Статистка**: В этом разделе приводится информация об изменении уровня шума в ADU и времени обработки изображения.

- **Укладка**: В этом разделе приводится информация о количестве сложенных кадров и общем времени экспозиции.

- **Конфигурация**: по умолчанию этот раздел не раскрыт. Когда это сделано, здесь приводится информация, проведена ли предобработка с использованием мастер-файлов, типе регистрации и укладки.

Примечание: Для использования мастер-файлов в ходе сессии укладки на лету, сначала необходимо уложить мастер-файлы. Когда это сделано, перед началом сессии, загрузите мастер-файлы на вкладке
Калибровка. В последующем эти файлы будут приняты во внимание в ходе укладки на лету и будут показаны в диалоге Конфигурация.

18.2 Укладка на лету (автономный режим)

Возможно использование укладки на лету из командной строки. Для этого необходимы всего 3 команды, которые объяснены ниже.

- Первая, `start_ls`, запускает сессию укладки на лету. Возможно указать этой команде темновые кадры и кадры плоского поля для калибровки изображений в ходе сессии.

**Командная строка Siril**

```
start_ls [-dark=filename] [-flat=filename] [-rotate] [-32bits]
```

Запускает сессию укладки на лету, опционально используя калибровочные файлы и ожидает входящие файлы от команды LIVESTACK пока не будет вызвана команда STOP_LS. По умолчанию используется регистрация по сдвигу и 16-битная обработка, из-за скорости. Эти настройки могут быть изменены с помощью аргументов -`rotate` (добавляет вращение) и -`32bits` (используется 32-битная разрядность)

Ссылки: `livestack`, `stop_ls`

- Команда `livestack` будет применена к каждому изображению, которое вы желаете сложить.

**Командная строка Siril**

```
livestack filename
```

Обрабатывает изображение в ходе укладки на лету. Возможна только после START_LS

Ссылка: `start_ls`
• Последняя команда, \textit{stop\_ls}, останавливает сессию укладки на лету.

<table>
<thead>
<tr>
<th>Код команда Siril</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{stop_ls}</td>
</tr>
</tbody>
</table>

Останавливает сессию укладки на лету. Возможно только после \texttt{START\_LS}

Ссылка: \texttt{start\_ls}
На этой странице содержится список всех команд, доступных в Siril 1.2.0.

Доступ к списку доступен по нажатию иконки.

Команды, обозначенные иконкой, могут использоваться в сценариях, а обозначенные иконкой — не могут.

**Совет:** Для всех команд, работающих с последовательностями, возможно заменить имя последовательности на точку ., если последовательность, которую необходимо обработать, уже загружена.

**Совет:** If you want to provide an argument that includes a string with spaces, for example a filename, you need to quote the entire argument not just the string. So for example you should use command "-filename=My File.fits", not command -filename=\"My File.fits\".

---

**addmax**

**Синтаксис:**

```
addmax filename
```

**Описание:**

Вычисляет новое изображение, объединяя изображение, загруженное в память, с изображением filename. Значение каждого пикселя в новом изображении определяется как максимальное значение в текущем изображении и в filename.

---

**asinh**
asinh [-human] stretch [offset]

Преобразует изображение, используя функцию arcsin, благодаря чему становятся видны тусклые объекты. Обязательный аргумент команды — stretch, обычно в диапазоне от 1 до 1000, определяет интенсивность растягивания. Точка чёрного может быть смешена с помощью аргумента offset в нормализованном значении пикселя в диапазоне [0, 1]. Наконец, аргумент -human позволяет использовать веса на основе относительной чувствительности человеческого глаза для вычисления светимости, применяемой при расчёте растягивания для каждого пикселя, а не просто средние значения пикселей каналов. Этот метод растягивания сохраняет светимость в цветовом пространстве L*a*b*

autoghs

autoghs [-linked] shadowsclip stretchamount [-b=] [-hp=] [-lp=]

Применение обобщенного гиперболического растягивания с точкой симметрии SP, определяемой как k. из медианы каждого канала (здесь значение shadowsclip является k и может быть отрицательным). По умолчанию точка симметрии и растягивание вычисляются поканально. Точка симметрии может быть рассчитана как среднее значение каналов при помощи опции -linked. Интенсивность растягивания D передаётся во втором обязательном аргументе. Используются следующие неявные значения: 13 для B, фокусирующее на диапазон яркости точки симметрии SP; 0.7 для HP и 0 для LP. Все значения могут быть изменены при помощи опций с соответствующими названиями

autostretch

autostretch [-linked] shadowsclip [targetbg]

Автоматически растягивает текущее загруженное изображение с различными параметрами для каждого канала (если каналы несвязаны), если только не передан аргумент -linked. Необязательные аргументы: shadowsclip — точка обрезки теней, выраженная в значениях от основного пика гистограммы (по умолчанию -2.8); targetbg — целевое значение фона, в диапазоне [0, 1] (по умолчанию 0.25), определяющее окончательную яркость изображения
Возвращает значение уровня фона изображения, загруженного в память

bgnoise Возвращает значение уровня фонового шума изображения, загруженного в память

binxy Computes the numerical binning of the in-memory image (sum of the pixels 2x2, 3x3..., like the analogic binning of CCD camera). If the optional argument -sum is passed, then the sum of pixels is computed, while it is the average when no optional argument is provided

boxselect Make a selection area in the currently loaded image with the arguments x, y, width and height, with x and y being the coordinates of the top left corner starting at (0, 0), and width and height, the size of the selection. The -clear argument deletes any selection area. If no argument is passed, the current selection is printed

Calibrate the sequence **sequencename** using bias, dark and flat given in argument.

For bias, a uniform level can be specified instead of an image, by entering a quoted expression starting with an = sign, such as -bias="=256" or -bias="=64*$OFFSET".

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with -cc= option.

You can use -cc=dark to detect hot and cold pixels from the masterdark (a masterdark must be given with the -dark= option), optionally followed by siglo and sighi for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied.

Alternatively, you can use -cc=bpm followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with a find_hot command on a masterdark.

It is possible to specify if images are CFA for cosmetic correction purposes with the option -cfa and also to demosaic images at the end of the process with -debayer.

The -fix_xtrans option is dedicated to X-Trans files by applying a correction on darks and biases to remove an ugly square pattern.

The -equalize_cfa option equalizes the mean intensity of RGB layers of the CFA flat master.

It is also possible to optimize the dark subtraction with -opt.

By default, frames marked as excluded will not be processed. The argument -all can be used to force processing of all frames even if marked as excluded.

The output sequence name starts with the prefix "pp_" unless otherwise specified with option -prefix=.

If -fitseq is provided, the output sequence will be a FITS sequence (single file)


Calibrate the image **imagename** using bias, dark and flat given in argument.

For bias, a uniform level can be specified instead of an image, by entering a quoted expression starting with an = sign, such as -bias="=256" or -bias="=64*$OFFSET".

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with -cc= option.

You can use -cc=dark to detect hot and cold pixels from the masterdark (a masterdark must be given
with the \texttt{-dark=} option, optionally followed by \texttt{siglo} and \texttt{sighi} for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied.

Alternatively, you can use \texttt{-cc=bpm} followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with \texttt{find_hot} command on a masterdark.

It is possible to specify if images are CFA for cosmetic correction purposes with the option \texttt{-cfa} and also to demosaic images at the end of the process with \texttt{-debayer}.

The \texttt{-fix_xtrans} option is dedicated to X-Trans files by applying a correction on darks and biases to remove an ugly square pattern.

The \texttt{-equalize_cfa} option equalizes the mean intensity of RGB layers of the CFA flat master.

It is also possible to optimize the dark subtraction with \texttt{-opt}.

Note that the command-line parser will not reject the argument \texttt{-pex}, which is used in preprocessing sequences, but it will be ignored.

The output filename starts with the prefix "pp_" unless otherwise specified with option \texttt{-prefix=}

\begin{verbatim}
capabilities
  capabilities

  catsearch star_name

  cd directory

  cd directory
\end{verbatim}

Sets the new current working directory.
The argument **directory** can contain the `~` token, expanded as the home directory, directories with spaces in the name can be protected using single or double quotes.

```
cdg

Возвращает координаты центра тяжести изображения
```

```
clahe

clahe cliplimit tileSize

Equalizes the histogram of an image using Contrast Limited Adaptive Histogram Equalization.

**cliplimit** sets the threshold for contrast limiting.
**tilesize** sets the size of grid for histogram equalization. Input image will be divided into equally sized rectangular tiles.
```

```
clear

Очищает содержимое журнала
```

```
clearstar

clearstar

Очистить всё звёзды, сохранённые в памяти и отображённые на экране
Корректно закрывает открытое изображение и открытую последовательность, если имеются

```
convert
```

Converts all images in a known format into Siril's FITS images.

The argument `basename` is the basename of the new sequence. For FITS images, Siril will try to make a symbolic link. If not possible, files will be copied.

The flags `fitseq` and `ser` can be used to specify an alternative output format, other than the default FITS.

The option `debayer` applies demosaicing to images. In this case no symbolic link are done.

```
```

```
convertraw
```

Converts DSLR RAW files into Siril's FITS images.

The argument `basename` is the basename of the new sequence. For FITS images, Siril will try to make a symbolic link. If not possible, files will be copied.

The flags `fitseq` and `ser` can be used to specify an alternative output format, other than the default FITS.

The option `debayer` applies demosaicing to images. In this case no symbolic link are done.

```
```

```
cosme
```
cosme [filename].lst

Применяет локальное среднее к набору пикселей на изображении, находящемся в памяти (косметическая коррекция). Координаты этих пикселей находятся в файле ASCII [файл .lst]. COSME адаптирована для коррекции остаточных горячих и холодных пикселей, сохранившихся после предварительной обработки.

cosme_cfa [filename].lst

Аналогична функции COSME, но применяемая к RAW-изображениям CFA.

Links: cosme

crop [x y width height]

Crops to a selected area of the loaded image.

If a selection is active, no further arguments are required. Otherwise, or in scripts, arguments have to be given, with x and y being the coordinates of the top left corner, and width and height the size of the selection. Alternatively, the selection can be made using the BOXSELECT command.

Links: boxselect

ddp level coef sigma

Performs a DDP (digital development processing) as described first by Kunihiko Okano. This implementation is the one described in IRIS.
It combines a linear distribution on low levels (below level) and a non-linear one on high levels. It uses a Gaussian filter of standard deviation sigma and multiplies the resulting image by coef. Typical values for sigma are within 0.7 and 2. The level argument should be in the range [0, 65535] for 16-bit images and may be given either in the range [0, 1] or [0, 65535] for 32-bit images in which case it will be scaled automatically.

```
denoise [-nocosmetic] [-mod=m] [-vst | -da3d | -sos=n [-rho=r]] [-indep]
```

Denoises the image using the non-local Bayesian algorithm described by Lebrun, Buades and Morel.

It is strongly recommended to apply cosmetic correction to remove salt and pepper noise before running denoise, and by default this command will apply cosmetic correction automatically. However, if this has already been carried out earlier in the workflow it may be disabled here using the optional command -nocosmetic.

An optional argument -mod=m may be given, where 0 <= m <= 1. The output pixel is computed as:

\[ \text{out} = m \times d + (1 - m) \times \text{in} \]

where \( d \) is the denoised pixel value. A modulation value of 1 will apply no modulation. If the parameter is omitted, it defaults to 1.

The optional argument -vst can be used to apply the generalised Anscombe variance stabilising transform prior to NL-Bayes. This is useful with photon-starved images such as single subs, where the noise follows a Poisson or Poisson-Gaussian distribution rather than being primarily Gaussian. It cannot be used in conjunction with DA3D or SOS, and for denoising stacked images it is usually not beneficial.

The optional argument -da3d can be used to enable Data-Adaptive Dual Domain Denoising (DA3D) as a final stage denoising algorithm. This uses the output of BM3D as a guide image to refine the denoising. It improves detail and reduces staircasing artefacts.

The optional argument -sos=n can be used to enable Strengthen-Operate-Subtract (SOS) iterative denoise boosting, with the number of iterations specified by n. In particular, this booster may produce better results if the un-boosted NL-Bayes algorithm produces artefacts in background areas. If both -da3d and -sos=n are specified, the last to be specified will apply.

The optional argument -rho=r may be specified, where 0 < r < 1. This is used by the SOS booster to determine the amount of noisy image added in to the intermediate result between each iteration. If -sos=n is not specified then the parameter is ignored.

The default is not to apply DA3D or SOS, as the improvement in denoising is usually relatively small and these techniques requires additional processing time.

In very rare cases, blocky coloured artefacts may be found in the output when denoising colour images. The optional argument -indep can be used to prevent this by denoising each channel separately. This is slower but will eliminate artefacts.
dir

Отображает список файлов и директорий в рабочей директории

This command is only available on Microsoft Windows, for the equivalent command on Linux and MacOS, see ls.

dumpheader

Отображает заголовок FITS

entropy

Вычисляет энтропию открытого изображения на отображаемом слое. При наличии выделения энтропия вычисляется для выделенной области или для всего изображения при отсутствии выделения. Энтропия это один из способов измерения шума или деталей на изображении

exit

Выход из приложения
extract NbPlans

Extracts **NbPlans** planes of wavelet domain

**extract_Green**

**extract_Green**

Extracts green signal from a CFA image. The output file name starts with the prefix "Green_"

**extract_Ha**

**extract_Ha**

Extracts Ha signal from a CFA image. The output file name starts with the prefix "Ha_"

**extract_HaOIII**

**extract_HaOIII** [-resample=

Extracts Ha and OIII signals from a CFA image. The output files names start with the prefix "Ha_" and "OIII_"

The optional argument **-resample={ha|oiii}** sets whether to upsample the Ha image or downsample the OIII image. If this argument is not provided, no resampling will be carried out and the OIII image will have twice the height and width of the Ha image
fdiv

\texttt{fdiv filename scalar}

Divides the image in memory by the image given in argument. The resulting image is multiplied by the value of the \texttt{scalar} argument. See also IDIV

Links: \textit{idiv}

ffill

\texttt{ffill value [x y width height]}

Аналогична команде FILL, но выполняет симметричную заливку области, определёной с помощью мыши. Используется для обработки изображения в домене Фурье (FFT)

Links: \textit{fill}

fftd

\texttt{fftd modulus phase}

Applies a Fast Fourier Transform to the image loaded in memory. \texttt{Modulus} and \texttt{phase} given in argument are saved in FITS files

ftti

\texttt{ftti modulus phase}

Retrieves corrected image applying an inverse transformation. The \texttt{modulus} and \texttt{phase} used are the files given in argument
fill value [x y width height]

Fills the whole current image (or selection) with pixels having the value intensity expressed in ADU

find_cosme cold_sigma hot_sigma

Применяет автоматическое определение холодных и горячих пикселей с использованием порогов, указанных в аргументах

find_cosme_cfa cold_sigma hot_sigma

Аналогична команде FIND_COSME, но для монохромных изображений CFA

Links: find_cosme

find_hot filename cold_sigma hot_sigma

Saves a list file filename (text format) in the working directory which contains the coordinates of the pixels which have an intensity hot_sigma times higher and cold_sigma lower than standard deviation. We generally use this command on a master-dark file

Lines P x y type will fix the pixel at coordinates (x, y) type is an optional character (C or H) specifying to Siril if the current pixel is cold or hot. This line is created by the command FIND_HOT but you also can add some lines manually:

Lines C x 0 type will fix the bad column at coordinates x.
Lines L y 0 type will fix the bad line at coordinates y.

```
findstar [-out=] [-layer=] [-maxstars=]
```

Detects stars in the currently loaded image, having a level greater than a threshold computed by Siril. After that, a PSF is applied and Siril rejects all detected structures that don't fulfill a set of prescribed detection criteria, that can be tuned with command SETFINDSTAR. Finally, a circle is drawn around detected stars.

Optional parameter -out= enables to save the results to the given path. Option -layer= specifies the layer onto which the detection is performed (for color images only). You can also limit the max number of stars detected by passing a value to option -maxstars=.

См. также команду CLEARSTAR

Links: psf, setfindstar, clearstar

```
fix_xtrans
```

Fixes the Fujifilm X-Trans Auto Focus pixels.

Indeed, because of the phase detection auto focus system, the photosites used for auto focus get a little less light than the surrounding photosites. The camera compensates for this and increases the values from these specific photosites giving a visible square in the middle of the dark/bias frames

```
fixbanding amount sigma [-vertical]
```

Tries to remove the canon banding.
Argument **amount** defines the amount of correction. **Sigma** defines a protection level of the algorithm, higher sigma gives higher protection. **-vertical** option enables to perform vertical banding removal.

```plaintext
fmedian ksize modulation
```

Performs a median filter of size **ksize x ksize** (ksize MUST be odd) to the original image with a modulation parameter **modulation**.

The output pixel is computed as: \( \text{out} = \text{mod} \times m + (1 \mod) \times \text{in} \), where \( m \) is the median-filtered pixel value. A modulation's value of 1 will apply no modulation.

```plaintext
fmul scalar
```

Multiplies the loaded image by the **scalar** given in argument.

```plaintext
gauss sigma
```

Performs a Gaussian filter with the given **sigma**.

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

get
Получает значение переменной, используя её имя или отображает все переменные с помощью -a (имя и значение) или -A (подробный список)

```bash
getref
getref sequencename
```

Выводит информацию об опорном изображении в последовательности, переданной в аргументе.
Индекс первого изображения — 0

```bash
ght
ght -D= [-B=] [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] [channels]
```

Generalised hyperbolic stretch based on the work of the ghsastro.co.uk team.

The argument **-D**= defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

- **B** defines the intensity of the stretch near the focal point, between -5 and 15;
- **LP** defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;
- **SP** defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;
- **HP** defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted B, LP and SP default to 0.0 and HP defaults to 1.0.

An optional argument (either **-human**, **-even** or **-independent**) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument **-sat** specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

Optionally the parameter **[channels]** may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels

```bash
grey_flat
```

Пользовательский интерфейс:

Имя: Siril, Выпуск 1.2.0

**Содержание:**

- Получение значения переменной
- Вывод информации о последовательности изображений
- Generalised hyperbolic stretch
- Модификация характеристик
- Выбор каналов для улучшения изображения
Выравнивает среднюю интенсивность слоёв RGB в изображении CFA

**help**

```
help [command]
```

Отображает список доступных команд или помощь для одной команды

**histo**

```
histo channel (channel=0, 1, 2 with 0: red, 1: green, 2: blue)
```

Вычисляет гистограмму указанного канала изображения, находящегося памяти и создает файл histo_[channel name].dat в рабочей директории

**iadd**

```
iadd filename
```

Adds the image in memory to the image **filename** given in argument. Result will be in 32 bits per channel if allowed in the preferences

**idiv**

```
idiv filename
```

Divides the image in memory by the image **filename** given in argument. Result will be in 32 bits per channel if allowed in the preferences. See also FDIV

Links: **fdiv**
imul

*imul filename*

Multiplies the image in memory by the image *filename* given in argument. Result will be in 32 bits per channel if allowed in the preferences.

inspector

*inspector*

Создаёт из текущего изображения мозаику из 9 изображений, отображающую углы и центр изображения для более тщательного изучения.

invght

*invght -D= [-B=] [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] [channels]*

Inverse generalised hyperbolic stretch based on the work of the ghsastro.co.uk team.

The argument *-D=* defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

- **B** defines the intensity of the stretch near the focal point, between -5 and 15;
- **LP** defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;
- **SP** defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;
- **HP** defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted B, LP and SP default to 0.0 and HP defaults to 1.0.

An optional argument (either *-human*, *-even* or *-independent*) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument *-sat* specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

Optionally the parameter *[channels]* may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.
invmodasinh defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

**LP** defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;

**SP** defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;

**HP** defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted LP and SP default to 0.0 and HP defaults to 1.0.

An optional argument (either `-human`, `-even` or `-independent`) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument `sat` specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

Optionally the parameter `[channels]` may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.
jsonmetadata

```
jsonmetadata FITS_file [-stats_from_loaded] [-nostats] [-out=]
```

Dumps metadata and statistics of the currently loaded image in JSON form. The file name is required, even if the image is already loaded. Image data may not be read from the file if it is the current loaded image and if the `-stats_from_loaded` option is passed. Statistics can be disabled by providing the `-nostats` option. A file containing the JSON data is created with default file name '$(FITS_file_without_ext).json' and can be changed with the `-out=` option.

light_curve

```
light_curve sequencename channel [-autoring] { -at=x,y | -wcs=ra,dec } { -refat=x,y | -refwcs=ra,dec } ...
light_curve sequencename channel [-autoring] -ninastars=file
```

Analyse several stars with aperture photometry in a sequence of images and produce a light curve for one, calibrated by the others. The first coordinates (in degrees) are for the star whose light will be plotted, the others for the reference stars. Alternatively, a list of target and reference stars can be passed in the format of the NINA exolpanet plugin star list, with the `-ninastars=` option. Siril will verify that all reference stars can be used before actually using them. A data file is created in the current directory named light_curve.dat, gnuplot plots the result to a PNG image if available. The ring radii for aperture photometry can either be configured in the settings or set to a factor of the reference image's FWHM if `-autoring` is passed.

linear_match

```
linear_match reference low high
```

Computes a linear function between a reference image and the image in memory. The function is then applied to the current image to match it to the reference one. The algorithm will ignore all reference pixels whose values are outside of the [low, high] range.
link

**link basename [-start=index] [-out=]**

Links all FITS images in the working directory with the *basename* given in argument.

If no symbolic links could be created, files are copied. It is possible to convert files in another directory with the *-out=* option.

linstretch

**linstretch -BP= [-sat] [channels]**

Stretches the image linearly to a new black point BP.

There is one mandatory argument:

**BP** provides the new black point to stretch to.

The argument **[channels]** may optionally be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

Optionally the parameter **-sat** may be used to apply the linear stretch to the image saturation channel. This argument only works if all channels are selected.

livestack

**livestack filename**

Обрабатывает изображение в ходе укладки на лету. Возможна только после START_LS

Ссылка: *start_ls*
**load filename[.ext]**

Loads the image *filename*

It first attempts to load *filename*, then *filename*.fit, finally *filename*.fits and finally all supported formats, aborting if none of these are found.

This scheme is applicable to every Siril command that involves reading files. Fits headers MIPS-HI and MIPS-LO are read and their values given to the current viewing levels.

Writing a known extension .ext at the end of *filename* will load specifically the image *filename*.ext: this is used when numerous files have the same name but not the same extension.

**log**

**ls**

This command is only available on Unix-like Systems, for the equivalent command on Microsoft Windows, see **dir**.

**makepsf**

```
makepsf clear
makepsf load filename
makepsf save [filename]
makepsf stars [-sym] [-ks=} [-savepsf=]
```
Generates a PSF for use with deconvolution. One of the following must be given as the first argument: `clear` (clears the existing PSF), `load` (loads a PSF from a file), `save` (saves the current PSF), `blind` (blind estimate of the PSF), `stars` (generates a PSF based on measured stars from the image) or `manual` (generates a PSF manually based on a function and parameters).

No additional arguments are required when using the `clear` argument.

To load a previously saved PSF the `load` argument requires the PSF `filename` as a second argument. This may be in any format that Siril has been compiled with support for, but it must be square and should ideally be odd.

To save a previously generated PSF the argument `save` is used. Optionally, a filename may be provided (this must have one of the extensions ".fit", ".fits", ".fts" or ".tif") but if none is provided the PSF will be named based on the name of the open file or sequence.

For `blind`, the following optional arguments may be provided: `-l0` uses the l0 descent method, `-si` uses the spectral irregularity method, `-multiscale` configures the l0 method to do a multi-scale PSF estimate, `-lambda=` provides the regularization constant.

For PSF from detected `stars` the only optional parameter is `-sym`, which configures the PSF to be symmetric.

For a `manual` PSF, one of `-gaussian`, `-moffat`, `-disc` or `-airy` can be provided to specify the PSF function, Gaussian by default. For Gaussian or Moffat PSFs the optional arguments `-fwhm=`, `-angle=` and `-ratio=` may be provided. For Moffat PSFs the optional argument `-beta=` may also be provided. If these values are omitted, they default to the same values as in the deconvolution dialog. For disc PSFs only the argument `-fwhm=` is required, which for this function is used to set the `diameter` of the PSF. For Airy PSFs the following arguments may be provided: `-dia=` (sets the telescope diameter), `-fl=` (sets the telescope focal length), `-wl=` (sets the wavelength to calculate the Airy diffraction pattern for), `-pixelsize=` (sets the sensor pixel size), `-obstruct=` (sets the central obstruction as a percentage of the overall aperture area). If these parameters are not provided, wavelength will default to 525nm and central obstruction will default to 0%. Siril will attempt to read the others from the open image, but some imaging software may not provide all of them in which case you will get bad results, and note the metadata may not be populated for SER format videos. You will learn from experience which are safe to omit for your particular imaging setup.

For any of the above PSF generation options the optional argument `-ks=` may be provided to set the PSF dimension, and the optional argument `-savepsf=filename` may be used to save the generated PSF: a filename must be provided and the same filename extension requirements apply as for `makepsf save` `filename`.

Links: `psf`
merge

merge sequence1 sequence2 [sequence3 ...] output_sequence

Слияние нескольких последовательностей в одну

merge_cfa

merge_cfa file_CFA0 file_CFA1 file_CFA2 file_CFA3 bayerpattern

Builds a Bayer masked colour image from 4 separate images containing the data from Bayer subchannels CFA0, CFA1, CFA2 and CFA3. (The corresponding command to split the CFA pattern into subchannels is split_cfa.) This function can be used as part of a workflow applying some processing to the individual Bayer subchannels prior to demosaicing. The fifth parameter bayerpattern specifies the Bayer matrix pattern to recreate: bayerpattern should be one of 'RGGB', 'BGGR', 'GRBG' or 'GBRG'

mirrorx

mirrorx [-bottomup]

Flips the image about the horizontal axis. Option -bottomup will only flip it if it's not already bottom-up

mirrorx_single

mirrorx_single image

Переворачивает изображение по горизонтали только при необходимости (если изображение уже не находится в ориентации снизу вверх). Принимает название файла как аргумент, позволяя не читать полностью данные изображения, если необходимость переворота отсутствует. После переворота изображение будет перезаписано
modasinh -D= [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] [channels]

Modified arcsinh stretch based on the work of the ghsastro.co.uk team.

The argument -D= defines the strength of the stretch, between 0 and 10. This is the only mandatory argument. The following optional arguments further tailor the stretch:

- **LP** defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail;
- **SP** defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense;
- **HP** defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted LP and SP default to 0.0 and HP defaults to 1.0.

An optional argument (either -human, -even or -independent) can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument -sat specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

Optionally the parameter [channels] may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

mtf low mid high [channels]

Applies midtones transfer function to the current loaded image.

Three parameters are needed, low, midtones and high where midtones balance parameter defines a nonlinear histogram stretch in the [0,1] range.

Optionally the parameter [channels] may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.
Изменяет значения пикселей загруженного в настоящий момент изображения на отрицательное представление, например, 1-значение для 32 бит, 65535-значение для 16 бит. Режим просмотра при этом не изменяется.

Создает новое изображение размером width x height.

Создает новое изображение размером width x height.

Нет данных о том, каким образом это поле может быть использовано в контексте программы Siril.

Отображает звезды из локального каталога по умолчанию, если изображение было получено с помощью GUI. Можете указать альтернативный каталог с помощью параметра -catalog, принимая значения tycho2, nomad, gaia, ppmxl, brightstars, apass. Звезды, не имеющие информации B-V, будут отображаться по умолчанию, но их можно исключить с помощью параметра -photo.

Заменяет нулевые значения на level значения. Это полезно перед операцией idiv или fdiv.
**offset**

**offset value**

Adds the constant **value** (specified in ADU) to the current image. This constant can take a negative value.

In 16-bit mode, values of pixels that fall outside of $[0, 65535]$ are clipped. In 32-bit mode, no clipping occurs.

**parse**

**parse str [-r]**

 Parses the string **str** using the information contained in the header of the image currently loaded. Main purpose of this command is to debug path parsing of header keys which can be used in other commands. 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.

If **str** starts with **$def** prefix, it will be recognized as a reserved keyword and looked for in the strings stored in gui_prepro.dark_lib, gui_prepro.flat_lib, gui_prepro.bias_lib or gui_prepro.stack_default for **$defdark**, **$defflat**, **$defbias** or **$defstack** respectively.

The keyword **$seqname$** can also be used when a sequence is loaded.

**pcc**


Run the Photometric Color Correction on the loaded image.

If the image has already been plate solved, the PCC can reuse the astrometric solution, otherwise, or if WCS or other image metadata is erroneous or missing, arguments for the plate solving must be passed: the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space.

focal length and pixel size can be passed with **-focal=** (in mm) and **-pixelsize=** (in microns), overriding values from image and settings.

you can force the plate solving to be remade using the **-platesolve** flag.
Unless `-noflip` is specified and the image is detected as being upside-down, the image will be flipped if a plate solving is run.

For faster star detection in big images, downsampling the image is possible with `-downscale`.

The limit magnitude of stars used for plate solving and PCC is automatically computed from the size of the field of view, but can be altered by passing a `+offset` or `-offset` value to `-limitmag=`, or simply an absolute positive value for the limit magnitude.

The star catalog used is NOMAD by default, it can be changed by providing `-catalog=apass`. If installed locally, the remote NOMAD can be forced by providing `-catalog=nomad`

Links: nomad

Plate solve the loaded image.

If the image has already been plate solved nothing will be done, unless the `-platesolve` argument is passed to force a new solve. If WCS or other image metadata is erroneous or missing, arguments must be passed: the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space (not mandatory for astrometry.net).

focal length and pixel size can be passed with `-focal=` (in mm) and `-pixelsize=` (in microns), overriding values from image and settings.

Unless `-noflip` is specified, if the image is detected as being upside-down, it will be flipped.

For faster star detection in big images, downsampling the image is possible with `-downscale`.

Images can be either plate solved by Siril using a star catalogue and the global registration algorithm or by astrometry.net's local solve-field command (enabled with `-localasnet`).

The following options apply to Siril's plate solve only.

The limit magnitude of stars used for plate solving is automatically computed from the size of the field of view, but can be altered by passing a `+offset` or `-offset` value to `-limitmag=`, or simply an absolute positive value for the limit magnitude.

The choice of the star catalog is automatic unless the `-catalog=` option is passed: if local catalogs are installed, they are used, otherwise the choice is based on the field of view and limit magnitude. If the option is passed, it forces the use of the remote catalog given in argument, with possible values: tycho2, nomad, gaia, ppmxl, brightstars, apass
pm "expression" [-rescale [low] [high]]

This command evaluates the expression given in argument as in PixelMath tool. The full expression must be between double quotes and variables (that are image names, without extension, located in the working directory in that case) must be surrounded by the token $, e.g. "$image1$ * 0.5 + $image2$ * 0.5". A maximum of 10 images can be used in the expression. Image can be rescaled with the option -rescale followed by low and high values in the range [0, 1]. If no low and high values are provided, default values are set to 0 and 1.


Calibrate the sequence sequencename using bias, dark and flat given in argument.

For bias, a uniform level can be specified instead of an image, by entering a quoted expression starting with an = sign, such as -bias="=256" or -bias="=64*$OFFSET".

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with -cc= option.
You can use -cc=dark to detect hot and cold pixels from the masterdark (a masterdark must be given with the -dark= option), optionally followed by siglo and sighi for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied.
Alternatively, you can use -cc=bpm followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with a find_hot command on a masterdark.

It is possible to specify if images are CFA for cosmetic correction purposes with the option -cfa and also to demosaic images at the end of the process with -debayer.
The -fix_xtrans option is dedicated to X-Trans files by applying a correction on darks and biases to remove an ugly square pattern.
The -equalize_cfa option equalizes the mean intensity of RGB layers of the CFA flat master.
It is also possible to optimize the dark subtraction with -opt.
By default, frames marked as excluded will not be processed. The argument -all can be used to force processing of all frames even if marked as excluded.
The output sequence name starts with the prefix "pp_" unless otherwise specified with option -prefix=.
If -fitseq is provided, the output sequence will be a FITS sequence (single file).
This command is now deprecated: CALIBRATE should be used instead.

Links: *calibrate*

```
```

Calibrate the image `imagename` using bias, dark and flat given in argument.

For bias, a uniform level can be specified instead of an image, by entering a quoted expression starting with an `=` sign, such as `-bias="=256"` or `-bias="=64*$OFFSET"`.

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with `-cc=` option.

You can use `-cc=dark` to detect hot and cold pixels from the masterdark (a masterdark must be given with the `-dark=` option), optionally followed by `siglo` and `sighi` for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied.

Alternatively, you can use `-cc=bpm` followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with a `find_hot` command on a masterdark.

It is possible to specify if images are CFA for cosmetic correction purposes with the option `-cfa` and also to demosaic images at the end of the process with `-debayer`.

The `-fix_xtrans` option is dedicated to X-Trans files by applying a correction on darks and biases to remove an ugly square pattern.

The `-equalize_cfa` option equalizes the mean intensity of RGB layers of the CFA flat master.

It is also possible to optimize the dark subtraction with `-opt`.

Note that the command-line parser will not reject the argument `-pex`, which is used in preprocessing sequences, but it will be ignored.

The output filename starts with the prefix "pp_" unless otherwise specified with option `-prefix=`

This command is now deprecated: CALIBRATE_SINGLE should be used instead.

Links: *calibrate_single*
psf [channel]

Performs a PSF (Point Spread Function) on the selected star and display the results. If provided, the **channel** argument selects the image channel on which the star will be analyzed. It can be omitted for monochrome images or when run from the GUI with one of the channels active in the view.

```
```

Finds and optionally performs geometric transforms on images of the sequence given in argument so that they may be superimposed on the reference image. Using stars for registration, this algorithm only works with deep sky images. Star detection options can be changed using SETFINDSTAR or the Dynamic PSF dialog. The detection is done on the green layer for colour images, unless specified by the `-layer=` option with an argument ranging from 0 to 2 for red to blue.

The `-2pass` and `-noout` options will only compute the transforms but not generate the transformed images, `-2pass` adds a preliminary pass to the algorithm to find a good reference image before computing the transforms. `-nostarlist` disables saving the star lists to disk.

The option `-transf=` specifies the use of either **shift**, **similarity**, **affine** or **homography** (default) transformations respectively.

The option `-drizzle` activates the sub-pixel stacking by up-scaling by 2 the generated images.

The option `-minpairs=` will specify the minimum number of star pairs a frame must have with the reference frame, otherwise the frame will be dropped and excluded from the sequence.

The option `-maxstars=` will specify the maximum number of star to find within each frame (must be between 100 and 2000). With more stars, a more accurate registration can be computed, but will take more time to run.

The pixel interpolation method can be specified with the `-interp=` argument followed by one of the methods in the list `[no, ne, cu, li]`. If `none` is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the `-noclamp` argument.

All images of the sequence will be registered unless the option `-selected` is passed, in that case the excluded images will not be processed.

If created, the output sequence name starts with the prefix "r_" unless otherwise specified with `-prefix=` option.

Links: setfindstar, psf
Пересматривает директории со сценариями и обновляет меню сценариев

Возвращает ошибку, если версия Siril старше, чем указанная в аргументе

Resamples image, either with a factor factor or for the target width or height provided by either of -width= or -height= . This is generally used to resize images, a factor of 0.5 divides size by 2. In the graphical user interface, we can see that several interpolation algorithms are proposed.

The pixel interpolation method can be specified with the -interp= argument followed by one of the methods in the list no[ne], ne[arest], cu[bic], la[nzos4], li[near], ar[ea]. If none is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the -noclamp argument

rgbcomp

rgbcomp red green blue [-out=result_filename]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename]
Создаёт композитное RGB изображение, используя три независимых изображения, или LRGB композитное изображение, используя дополнительное яркостное изображение и три монохромных изображения или цветное изображение. Итоговое изображение будет иметь название composed_rgb.fit или composed_lrgb.fit, если в необязательном аргументе не указано другое название.

**rgradient**

```bash
rgradient xc yc dR dalpha
```

Creates two images, with a radial shift (dR in pixels) and a rotational shift (dalpha in degrees) with respect to the point (xc, yc).

Between these two images, the shifts have the same amplitude, but an opposite sign. The two images are then added to create the final image. This process is also called Larson Sekanina filter.

**rl**

```bash
```

Restores an image using the Richardson-Lucy method.

Optionally, a PSF may be loaded using the argument `-loadpsf=filename`.

The number of iterations is provide by `-iters` (the default is 10).

The type of regularization can be set with `-tv` for Total Variation, or `-fh` for the Frobenius norm of the Hessian matrix (the default is none) and `-alpha=` provides the regularization strength (lower value = more regularization, default = 3000).

By default the gradient descent method is used with a default step size of 0.0005, however the multiplicative method may be specified with `-mul`.

The stopping criterion may be activated by specifying a stopping limit with `-stop=`

Links: [psf](#)
rmgreen

```
rmgreen [-nopreserve] [type] [amount]
```

Applies a chromatic noise reduction filter. It removes green tint in the current image. This filter is based on PixInsight’s SCNR and it is also the same filter used by HLVG plugin in Photoshop. Lightness is preserved by default but this can be disabled with the -nopreserve switch.

**Type** can take values 0 for average neutral, 1 for maximum neutral, 2 for maximum mask, 3 for additive mask, defaulting to 0. The last two can take an **amount** argument, a value between 0 and 1, defaulting to 1.

rotate

```
rotate degree [-nocrop] [-interp=] [-noclamp]
```

Rotates the image by an angle of **degree** value. The option -nocrop can be added to avoid cropping to the image size (black borders will be added).

Note: if a selection is active, i.e. by using a command `boxselect` before `rotate`, the resulting image will be a rotated crop. In this particular case, the option -nocrop will be ignored if passed.

The pixel interpolation method can be specified with the -interp= argument followed by one of the methods in the list no[he], ne[arest], cu|bic, la|nczos4, li|near, ar|ea}. If none is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the -nclamp argument.

rotatePi

```
rotatePi
```

Вращает изображение на 180° вокруг центра. Это эквивалентно команде "ROTATE 180" или "ROTATE -180"

Links: rotate
**satu**

```
satu amount [background_factor [hue_range_index]]
```

Enhances the color saturation of the loaded image. Try iteratively to obtain best results. 

amount can be a positive number to increase color saturation, negative to decrease it, 0 would do nothing, 1 would increase it by 100%

background_factor is a factor to (median + sigma) used to set a threshold for which only pixels above it would be modified. This allows background noise to not be color saturated, if chosen carefully. Defaults to 1. Setting 0 disables the threshold.

hue_range_index can be [0, 6], meaning: 0 for pink to orange, 1 for orange to yellow, 2 for yellow to cyan, 3 for cyan, 4 for cyan to magenta, 5 for magenta to pink, 6 for all (default)

**save**

```
save filename
```

Saves current image to filename.fit (or .fits, depending on your preferences, see SETEXT). Fits headers MIPS-HI and MIPS-LO are added with values corresponding to the current viewing levels

Links: setext

**savebmp**

```
savebmp filename
```

Saves current image under the form of a bitmap file with 8-bit per channel: filename.bmp (BMP 24-bit)

**savejpg**

```
savejpg filename [quality]
```

Saves current image into a JPG file: filename.jpg.
You have the possibility to adjust the quality of the compression. A value 100 for `quality` parameter offers best fidelity while a low value increases the compression ratio. If no value is specified, a default value of 100 is applied.

`savepng` saves current image into a PNG file: `filename.png`, with 16 bits per channel if the loaded image is 16 or 32 bits, and 8 bits per channel if the loaded image is 8 bits.

`savepnm` saves current image under the form of a Netpbm file format with 16-bit per channel. The extension of the output will be `filename.ppm` for RGB image and `filename.pgm` for gray-level image.

`savetif` saves current image under the form of an uncompressed TIFF file with 16-bit per channel: `filename.tif`. The option `-astro` allows saving in Astro-tiff format, while `-deflate` enables compression.

`savetif32` saves current image under the form of a uncompressed TIFF file with 16-bit per channel: `filename.tif`. The option `-astro` allows saving in Astro-tiff format, while `-deflate` enables compression.
Same command as SAVETIF but the output file is saved in 32-bit per channel: filename.tif. The option -astro allows saving in Astro-tiff format, while -deflate enables compression

Links: savetif

```
savetif8 filename [-astro] [-deflate]
```

Same command as SAVETIF but the output file is saved in 8-bit per channel: filename.tif. The option -astro allows saving in Astro-tiff format, while -deflate enables compression

Links: savetif

```
sb [-loadpsf=] [-alpha=] [-iters=]
```

Restores an image using the Split Bregman method.

Optionally, a PSF may be loaded using the argument `-loadpsf=filename`.

The number of iterations is provide by `-iters` (the default is 1).

The regularization factor `-alpha=` provides the regularization strength (lower value = more regularization, default = 3000)

Links: psf

```
select sequencename from to
```

This command allows easy mass selection of images in the sequence sequencename (from from to to included)
seqapplyreg


Applies geometric transforms on images of the sequence given in argument so that they may be superimposed on the reference image, using registration data previously computed.

The output sequence name starts with the prefix "r_" unless otherwise specified with -prefix= option.

The option -drizzle activates up-scaling by 2 the images created in the transformed sequence.

The pixel interpolation method can be specified with the -interp= argument followed by one of the methods in the list no[ne], ne[arest], cu[bic], la[nzos4], li[near], ar[ea]. If none is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation. Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the -noclamp argument.

The registration is done on the first layer for which data exists for RGB images unless specified by -layer= option (0, 1 or 2 for R, G and B respectively).

Automatic framing of the output sequence can be specified using -framing= keyword followed by one of the methods in the list { current | min | max | cog }:
- **framing=max** (bounding box) adds a black border around each image as required so that no part of the image is cropped when registered.
- **framing=min** (common area) crops each image to the area it has in common with all images of the sequence.
- **framing=cog** determines the best framing position as the center of gravity (cog) of all the images.

Filtering out images:
Images to be registered can be selected based on some filters, like those selected or with best FWHM, with some of the -filter-* options.

With filtering being some of these in no particular order or number:
Best images from the sequence can be stacked by using the filtering arguments. Each of these arguments can remove bad images based on a property their name contains, taken from the registration data, with either of the three types of argument values:
- 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.

It is also possible to use manually selected images, either previously from the GUI or with the select or unselect commands, using the -filter-included argument.

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

This command clears selection, registration and/or statistics data stored in `sequencename`. You can specify to clear only registration, statistics and/or selection with `-reg`, `-stat` and `-sel` options respectively. All are cleared if no option is passed.

```
seqcosme sequencename [filename].lst [-prefix=]
```

Same command as COSME but for the the sequence `sequencename`. The output sequence name starts with the prefix "cosme_" unless otherwise specified with option `-prefix=`.

Links: [cosme](#)

```
seqcosme_cfa sequencename [filename].lst [-prefix=]
```

Same command as COSME_CFA but for the the sequence `sequencename`. 
The output sequence name starts with the prefix "cosme_" unless otherwise specified with option -prefix=

Links: cosme_cfa

```
seqcrop sequencename x y width height [-prefix=]
```

Crops the sequence given in argument `sequencename`.

The crop selection is specified by the upper left corner position `x` and `y` and the selection `width` and `height`. The output sequence name starts with the prefix "cropped_" unless otherwise specified with -prefix= option

```
seqextract_Green sequencename [-prefix=]
```

Same command as EXTRACT_GREEN but for the sequence `sequencename`.

The output sequence name starts with the prefix "Green_" unless otherwise specified with option -prefix=

```
seqextract_Ha sequencename [-prefix=]
```

Same command as EXTRACT_HA but for the sequence `sequencename`.

The output sequence name starts with the prefix "Ha_" unless otherwise specified with option -prefix=
seqextract_HaOIII sequecnename [-resample=]

Same command as EXTRACT_HAOIII but for the sequence sequencename.

The output sequences names start with the prefixes "Ha_" and "OIII_"

The optional argument -resample={ha|oiii} sets whether to upsample the Ha image or downsample the OIII image. If this argument is not provided, no resampling will be carried out and the OIII image will have twice the height and width of the Ha image.

seqght

seqght sequence -D= [-B=} [-LP=} [-SP=} [-HP=} [-human | -even | -independent | -sat]} channels [-prefix=]

Analogична команде GHT, но имя последовательности должно быть указано в первом аргументе.

Links: ght

seqheader

seqheader sequencename keyword [-out=file.csv]

Выводит значение указанного ключевого слова заголовка FITS для всех изображений последовательности.

seqfind_cosme

seqfind_cosme sequencename cold_sigma hot_sigma [-prefix=]

Same command as FIND_COSME but for the sequence sequencename.
The output sequence name starts with the prefix "cc_" unless otherwise specified with \texttt{-prefix=} option.

Links: \textit{find\_cosme}

\begin{verbatim}
seqfind\_cosme\_cfa sequencename cold\_sigma hot\_sigma \texttt{[-prefix=]}
\end{verbatim}

Same command as FIND\_COSME\_CFA but for the sequence \texttt{sequencename}.

The output sequence name starts with the prefix "cc_" unless otherwise specified with \texttt{-prefix=} option.

Links: \textit{find\_cosme\_cfa}

\begin{verbatim}
seqfindstar sequencename \texttt{[-layer=]} \texttt{[-maxstars=]}
\end{verbatim}

Same command as FIND\_STAR but for the sequence \texttt{sequencename}.

The option \texttt{-out=} is not available for this process as all the star list files are saved with the default name \texttt{sequencename\_seqnb.lst}

Links: \textit{find\_star}

\begin{verbatim}
seqfixbanding sequencename amount sigma \texttt{[-prefix=]} \texttt{[-vertical]}
\end{verbatim}

Same command as FIX\_BANDING but for the sequence \texttt{sequencename}.

The output sequence name starts with the prefix "unband_" unless otherwise specified with \texttt{-prefix=} option.
\texttt{-vertical} option enables to perform vertical banding removal.
Links: fixbanding

seqinvght

```
seqinvght sequence -D= [-B=] [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] [channels] [-prefix=]
```

Same command as INVGHT but the sequence must be specified as the first argument. In addition, the optional argument `-prefix=` can be used to set a custom prefix.

Links: invght

seqinvmodasinh

```
seqinvmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-human | -even | -independent | -sat] [channels] [-prefix=]
```

Same command as INVMODASINH but the sequence must be specified as the first argument. In addition, the optional argument `-prefix=` can be used to set a custom prefix.

Links: invmodasinh

seqlinstretch

```
seqlinstretch sequence -BP= [channels] [-sat] [-prefix=]
```

Same command as LINSTRETCH but the sequence must be specified as the first argument. In addition, the optional argument `-prefix=` can be used to set a custom prefix.

Links: linstretch

seqmerge_cfa

```
```


seqmerge_cfa sequencename bayerpattern [-prefixin=] [-prefixout=]

Same command as MERGE_CFA but for the sequence **sequencename**.

The Bayer pattern to be reconstructed must be provided as the second argument as one of RGGB, BGGR, GBRG or GRBG.

The input filenames contain the identifying prefix "CFA_" and a number unless otherwise specified with **-prefixin=** option.

Note: all 4 sets of input files **must** be present and **must** be consistently named, the only difference being the number after the identifying prefix.

The output sequence name starts with the prefix "mCFA_" and a number unless otherwise specified with **-prefixout=** option.

Links: merge_cfa

seqmodasinh


Same command as INVMODASINH but the sequence must be specified as the first argument. In addition, the optional argument **-prefix=** can be used to set a custom prefix.

Links: invmodasinh

seqmtf

seqmtf sequencename low mid high [channels] [-prefix=]

Same command as MTF but for the sequence **sequencename**.

The output sequence name starts with the prefix "mtf_" unless otherwise specified with **-prefix=** option.

Optionally the parameter **[channels]** may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.
Links: mtf

seqpsf

seqpsf [sequencename channel { -at=x,y | -wcs=ra,dec }]

Same command as PSF but for a sequence.

Results are displayed in the plots tab if used from the GUI, else it is printed in the console in a form that can be used to produce brightness variation curves. For headless operation, arguments are mandatory and the center of the search box in pixels can be provided with the -at= argument

Links: psf

seqplatesolve

seqplatesolve sequencename [image_center_coords] [-noflip] [-platesolve] [-focal=} [-pixelsize=} [-limitmag=[+-]} [-catalog=} [-localasnet] [-downscale]}

Plate solve a sequence. A new sequence will be created with the prefix "ps_".
If images have already been plate solved, they will just be copied, unless the -platesolve argument is passed to force a new solve. If WCS or other image metadata are erroneous or missing, arguments must be passed:
the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space (not mandatory for astrometry.net).
focal length and pixel size can be passed with -focal= (in mm) and -pixelsize= (in microns), overriding values from images and settings.

Unless -noflip is specified, if images are detected as being upside-down, they will be flipped.
For faster star detection in big images, downsampling the image is possible with -downscale.

Images can be either plate solved by Siril using a star catalogue and the global registration algorithm or by astrometry.net's local solve-field command (enabled with -localasnet).
The following options apply to Siril's plate solve only.
The limit magnitude of stars used for plate solving is automatically computed from the size of the field of view, but can be altered by passing a +offset or -offset value to -limitmag=, or simply an absolute positive value for the limit magnitude.
The choice of the star catalog is automatic unless the \texttt{-catalog=} option is passed: if local catalogs are installed, they are used, otherwise the choice is based on the field of view and limit magnitude. If the option is passed, it forces the use of the remote catalog given in argument, with possible values: tycho2, nomad, gaia, ppmxl, brightstars, apass.

\begin{verbatim}
seql sequencename [-loadpsf=} [-alpha=} [-iters=} [-stop=} [-gdstep=} [-tv} [-fh} [-mul]
\end{verbatim}

The same as the \texttt{RL} command, but applies to a sequence which must be specified as the first argument

\texttt{Cсылка: } \texttt{rl}

\begin{verbatim}
sb sequencename [-loadpsf=} [-alpha=} [-iters=]
\end{verbatim}

The same as the \texttt{SB} command, but applies to a sequence which must be specified as the first argument

\texttt{Cсылка: } \texttt{sb}

\begin{verbatim}
seqsplit_cfa sequencename [-prefix=]
\end{verbatim}

Same command as SPLIT\_CFA but for the sequence \texttt{sequencename}.

The output sequences names start with the prefix "CFA_" and a number unless otherwise specified with \texttt{-prefix=} option

\texttt{Links: } \texttt{split\_cfa}
seqstarnet

seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]

This command calls Starnet++ to remove stars from the current sequence. The first argument must be the sequence from which to remove stars: all the other available arguments are the same as for the STARNET command.

Links: starnet

seqstat

seqstat sequencename output [option] [-cfa]

Same command as STAT for sequence sequencename.

The output is saved as a csv file given in second argument.
The optional parameter defines the number of statistical values computed: basic, main (default) or full (more detailed but longer to compute).
If -cfa is passed and the images are CFA, statistics are made on per-filter extractions.

Links: stat

seqsubsky

seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-prefix=]

Аналогична команде SUBSKY, но применяется последовательности название последовательности.
Dithering, required for low dynamic gradients, can be disabled with -nodither.

The output sequence name starts with the prefix "bkg_" unless otherwise specified with -prefix= option.

Ссылка: subsky
seqtilt

```
seqtilt sequencename
```

Same command as TILT but for the loaded sequence or the sequence `sequencename`.

It generally gives better result.

Links: `tilt`

sequnsetmag

```
sequnsetmag
```

Сбрасывает калибровку звёздной величины и опорную звезду для последовательности. См. SEQSETMAG

Links: `seqsetmag`

seqwiener

```
wiener sequencename [-loadpsf=] [-alpha=]
```

The same as the WIENER command, but applies to a sequence which must be specified as the first argument.

Ссылка: `wiener`

set

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

Обновляет значение настройки, используя имя её переменной с заданным значением или набором значений, используя существующий ini-файл с опцией `-import=`.
См. get для получения списка переменных или их значений

**set16bits**

**set16bits**

Отключает сохранение изображений с разрядность 32 бита на канал при обработке. Вместо этого используются 16 бит

**set32bits**

**set32bits**

Позволяет сохранение изображений с 32 битами на канал при обработке

**setcompress**

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

Определяет, сжимать изображения или нет.

0 означает, что сжатие выключено, а 1 включает сжатие.
Если сжатие включено, его тип должен быть явно указан опцией -type= ("rice", "gzip1", "gzip2"). Связанное со сжатием значение квантования должно быть в диапазоне [0, 256]. Например, "setcompress 1 -type=rice 16" устанавливает тип сжатия (rice) со значением квантования 16

**setcpu**

**setcpu number**

Defines the number of processing threads used for calculation.
Can be as high as the number of virtual threads existing on the system, which is the number of CPU cores or twice this number if hyperthreading (Intel HT) is available.

```
setext

setext extension
```

Устанавливает расширение, используемое и понимаемое последовательностями.

Аргумент `extension` может быть "fit", "fts" или "fits"

```
setfindstar

```

Defines stars detection parameters for FINDSTAR and REGISTER commands.

Passing no parameter lists the current values.
Passing `reset` resets all values to defaults. You can then still pass values after this keyword.

Configurable values:

- `-radius=` defines the radius of the initial search box and must be between 3 and 50.
- `-sigma=` defines the threshold above noise and must be greater or equal to 0.05.
- `-roundness=` defines minimum star roundness and must between 0 and 0.95. `-maxR` allows an upper bound to roundness to be set, to visualize only the areas where stars are significantly elongated, do not change for registration.
- `-minA` and `-maxA` define limits for the minimum and maximum amplitude of stars to keep, normalized between 0 and 1.
- `-focal=` defines the focal length of the telescope.
- `-pixelsize=` defines the pixel size of the sensor.
- `-gaussian` and `-moffat` configure the solver model to be used (Gaussian is the default). If Moffat is selected, `-minbeta=` defines the minimum value of beta for which candidate stars will be accepted and must be greater or equal to 0.0 and less than 10.0.
- `-convergence=` defines the number of iterations performed to fit PSF and should be set between 1 and 3 (more tolerant).
**-relax**—relaxes the checks that are done on star candidates to assess if they are stars or not, to allow objects not shaped like stars to still be accepted (off by default)

Links: *findstar, register, psf*

---

**setmag**

**setmag magnitude**

Calibrates the magnitude by selecting a star and giving the known apparent magnitude.

All PSF computations will return the calibrated apparent magnitude afterwards, instead of an apparent magnitude relative to ADU values.
To reset the magnitude constant see UNSETMAG

Links: *psf, unsetmag*

---

**seqsetmag**

**seqsetmag magnitude**

Same as SETMAG command but for the loaded sequence.

This command is only valid after having run SEQPSF or its graphical counterpart (select the area around a star and launch the PSF analysis for the sequence, it will appear in the graphs).
This command has the same goal as SETMAG but recomputes the reference magnitude for each image of the sequence where the reference star has been found.
When running the command, the last star that has been analysed will be considered as the reference star. Displaying the magnitude plot before typing the command makes it easy to understand.
To reset the reference star and magnitude offset, see SEQUNSETMAG

Links: *setmag, seqpsf, psf, sequnsetmag*
setmem ratio

Sets a new ratio of free memory on memory used for stacking.

**Ratio** value should be between 0.05 and 2, depending on other activities of the machine. A higher ratio should allow siril to stack faster, but setting the ratio of memory used for stacking above 1 will require the use of on-disk memory, which is very slow and unrecommended.


Gets or sets photometry settings, mostly used by SEQPSF. If arguments are provided, they will update the settings. None are mandatory, any can be provided, default values are shown in the command's syntax. At the end of the command, the active configuration will be printed. Aperture is dynamic unless forced, the **aperture** value from settings is not used if dynamic, FWHM is used instead.

Links: seqpsf

setref sequencename image_number

Устанавливает опорное изображение в последовательности, указанной в первом аргументе.

show [-clear] [{ -list=file.csv | [name] RA Dec }]

Shows a point on the loaded plate solved image using the temporary user annotations catalogue, based on its equatorial coordinates. The **-clear** option clears this catalogue first and can be used alone. Several points can be passed using a CSV file with the option **-file** containing a name, ra and dec. This is only available from the GUI of Siril.
solsys

solsys [-mag=20.0]

Search and display solar system objects in the current loaded and plate solved image. Use `-mag=` to change the limit magnitude which defaults to 20

This research has made use of IMCCE's SkyBoT VO tool

split

split file1 file2 file3 [-hsl | -hsv | -lab]

Splits the color image into three distinct files (one for each color) and save them in file1.fit, file2.fit and file3.fit files. A last argument can optionally be supplied, -hsl, -hsv or lab to perform an HSL, HSV or CieLAB extraction. If no option are provided, the extraction is of RGB type

split_cfa

split_cfa

Разделяет изображение CFA на четыре отдельных файла (по одному для каждого канала) и сохраняет их в файлы

stack

stack seqfilename
stack seqfilename { sum | min | max } [filtering] [-output_norm] [-out=filename]
stack seqfilename { med | median } [-nonorm, -norm=*] [filtering] [-fastnorm] [-rgb_equal] [-output_norm] [-out=filename]
stack seqfilename { rej | mean } [rejection type] [sigma_low sigma_high] [-rejmap[s]] [-nonorm, -norm=*] [filtering] [-fastnorm] [-weight_from_noise | -weight_from_nbstack | -weight_from_wfwhm | -weight_from_nbstars ] [-rgb_equal] [-output_norm] [-out=filename]
Stacks the `sequencename` sequence, using options.

Тип выбраковки:
The allowed types are: "sum", "max", "min", "med" (or "median") and "rej" (or "mean"). If no argument other than the sequence name is provided, sum stacking is assumed.

Stacking with rejection:
Types rej or mean require the use of additional arguments for rejection type and values. The rejection type is one of \{ n[one] | p[ercentile] | s[igma] | m[edian] | w[insorized] | l[inear] | g[eneralized] | m[a][d] \} for Percentile, Sigma, Median, Winsorized, Linear-Fit, Generalized Extreme Studentized Deviate Test or k-MAD clipping. If omitted, the default (Winsorized) is used.
The `sigma low` and `sigma high` parameters of rejection are mandatory unless `none` is selected.
Optionally, rejection maps can be created, showing where pixels were rejected in one (-rejmap) or two (-rejmaps, for low and high rejections) newly created images.

Normalization of input images:
For med|median and rej|mean stacking types, different types of normalization are allowed: -norm=add for additive, -norm=mul for multiplicative. Options -norm=addscale and -norm=mulscale apply same normalization but with scale operations. -nonorm is the option to disable normalization. Otherwise additive with scale method is applied by default.
-fastnorm option specifies to use faster estimators for location and scale than the default IKSS.
-rgb_equal will use normalization to equalize color image backgrounds, useful if pcc and unlinked autostretch will not be used.

Other options for rejection stacking:
-weight_from_noise is an option to add larger weights to frames with lower background noise.
-weight_from_nbstack weights input images based on how many images were used to create them, useful for live stacking.
-weight_from_nbstars or -weight_from_wfwhm weight input images based on number of stars or wFWHM computed during registration step.

Outputs:
Result image name can be set with the -out= option. Otherwise, it will be named as `sequencename_stacked.fit`.
-output_norm applies a normalization to rescale result in the \([0, 1]\) range (median and mean stacking only).

Filtering out images:
Images to be stacked can be selected based on some filters, like manual selection or best FWHM, with some of the -filter-* options.

With filtering being some of these in no particular order or number:

```bash
[-filter-fwhm=value[\%|k]] [-filter-wfwhm=value[\%|k]] [-filter-round=value[\%|k]] [-filter-bkg=value[\%|k]]
```

(continues on next page)
Best images from the sequence can be stacked by using the filtering arguments. Each of these arguments can remove bad images based on a property their name contains, taken from the registration data, with either of the three types of argument values:
- 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.
It is also possible to use manually selected images, either previously from the GUI or with the select or unselect commands, using the \texttt{-filter-included} argument.

\begin{verbatim}
stackall { sum | min | max } [filtering]
stackall { med | median } [-nonorm, norm=] [-filter-incl[uded]]
stackall { rej | mean } [rejection type] [sigma_low sigma_high] [-nonorm, norm=]
  [filtering] [ -weight_from_noise | -weight_from_wfwhm | -weight_from_nbstars | -weight_from_nbstack ] [-rgb_equal] [-out=filename]
\end{verbatim}

\begin{verbatim}
Открывает все последовательности в текущей директории и укладывает их с помощью опционально указанных типов укладки или фильтрации или с помощью суммирования. Для описания опций см. STACK

Links: \textit{stack}
\end{verbatim}

\begin{verbatim}
starnet [-stretch] [-upscale] [-stride=value] [-nostarmask]
\end{verbatim}

This command calls \texttt{StarNet} to remove stars from the current image.

\textbf{Prerequisite:} StarNet is an external program, with no affiliation with Siril, and must be installed correctly prior to first use of this command, with the path to its installation directory correctly set in Preferences / Miscellaneous. The directory must contain the Command Line Tool version (not GUI version which exists for Windows users).
The starless image is loaded on completion, and a star mask image is created in the working directory unless the optional parameter -nostarmask is provided.

Optionally, parameters may be passed to the command:
- 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.

More tips and tricks are available there

```
sstart_ls [-dark=filename] [-flat=filename] [-rotate] [-32bits]
```

Запускает сессию укладки на лету, опционально используя калибровочные файлы и ожидает входящие файлы от команды LIVESTACK пока не будет вызвана команда STOP_LS. По умолчанию используется регистрация по сдвигу и 16 битная обработка, из-за скорости. Эти настройки могут быть изменены с помощью аргументов -rotate (добавляет вращение) и -32bits (используется 32 битная разрядность)

Ссылки: livestack, stop_ls

```
stat [-cfa] [main]
```

Returns statistics of the current image, the basic list by default or the main list if main is passed. If a selection is made, statistics are computed within the selection. If -cfa is passed and the image is CFA, statistics are made on per-filter extractions

```
stop_ls
```

362 Глава 19. Команды
**stop_ls**

Останавливает сессию укладки на лету. Возможно только после START_LS

Ссылка: *start_ls*

**subsky**

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5]
```

Вычисляет синтетический градиент фона, используя либо полиномиальную функцию степени **degree** или радиально-базисную функцию (RBF), если указан аргумент `-rbf`, и вычитает градиент из изображения.

Количество и плотность образцов в горизонтальной строке и допуск для исключения ярких областей могут быть указаны опционально с помощью соответствующих аргументов. Допуск указывается в единицах медианного абсолютного отклонения (mad): медиана + допуск * mad.

Шум, необходимый для низких динамических градиентов, может быть подмешан с помощью аргумента `-dither`.

Для RBF так же доступен дополнительный параметр сглаживания

**synthstar**

```
synthstar
```

Synthstar fixes bad stars. No matter how much coma, tracking drift or other distortion your stars have, if Siril’s star finder routine can detect it, synthstar will fix it. To use intensive care, you may wish to manually detect all the stars you wish to fix. This can be done using the findstar console command or the Dynamic PSF dialog. If you have not run star detection, it will be run automatically with default settings.

For best results synthstar should be run before stretching.

The output of synthstar is a fully corrected synthetic star mask comprising perfectly round star PSFs (Moffat or Gaussian profiles depending on star saturation) computed to match the intensity, FWHM, hue and saturation measured for each star detected in the input image. This can then be recomposed with the starless image to produce an image with perfect stars.

No parameters are required for this command

Links: *psf*
threshlo

`threshlo level`

Replaces values below `level` with `level`

threshhi

`threshhi level`

Replaces values above `level` with `level`

tresh

`thresh lo hi`

Replaces values below `level` with `level`

tilt

`tilt [clear]`

Computes the sensor tilt as the fwhm difference between the best and worst corner truncated mean values. The `clear` option allows to clear the drawing.

unclipstars
unclipstars

Re-profiles clipped stars to desaturate them, scaling the output so that all pixel values are \( \leq 1.0 \)

unselect

**unselect sequencename from to**

 Allows easy mass unselection of images in the sequence `sequencename` (from `from` to `to` included). See SELECT

Links: `select`

unsetmag

**unsetmag**

Сбрасывает калибровку блеска к 0. См. SETMAG

Links: `setmag`

unsharp

**unsharp sigma multi**

Applies to the working image an unsharp mask with sigma `sigma` and coefficient `multi`

visu
**visu low high**

Displays an image with **low** and **high** as the low and high threshold

**wavelet**

**wavelet nbr_layers type**

Computes the wavelet transform on \((\text{nbr\_layers}=1...6)\) layer(s) using linear \((\text{type}=1)\) or bspline \((\text{type}=2)\) version of the 'a trous' algorithm. The result is stored in a file as a structure containing the layers, ready for weighted reconstruction with **wrecons**

Links: [wrecons](#)

**wiener**

**wiener [-loadpsf=] [-alpha=]**

Restores an image using the Wiener deconvolution method.

Optionally, a PSF may be loaded using the argument **-loadpsf=** **filename**.

The parameter **-alpha=** provides the Gaussian noise modelled regularization factor

Links: [psf](#)

**wrecons**

**wrecons c1 c2 c3 ...**

Reconstructs to current image from the layers previously computed with wavelets and weighted with coefficients \(c_1, c_2, ..., c_n\) according to the number of layers used for wavelet transform


[ConejeroPI] Juan Conejero, ImageIntegration, PixInsight Tutorial


Алфавитный указатель

A
admax, 307
asinh, 307
autoghs, 308
autostretch, 308

B
bg, 308
bgnoise, 309
binxy, 309
boxselect, 309

C
calibrate, 309
calibrate_single, 310
capabilities, 311
catsearch, 311
cd, 311
cdg, 312
clahe, 312
clear, 312
clearstar, 312
close, 313
convert, 313
convertraw, 313
cosme, 313
cosme_cfa, 314
crop, 314

ddpc, 314
denoise, 315
dir, 316
dumpheader, 316

E
entropy, 316
exit, 316
extract, 317
extract_Green, 317
extract_Ha, 317
extract_HaOIII, 317
fdiv, 318
ffill, 318
fftd, 318
fft1, 318
fill, 319
find_cosme, 319
find_cosme_cfa, 319
find_hot, 319
findstar, 320
fix_xtrans, 320
fixbanding, 320
fmedian, 321
fmul, 321
gauss, 321
get, 321
getref, 322
ght, 322
grey_flat, 322
help, 323
histo, 323
iadd, 323
idiv, 323
imul, 324
inspector, 324
invght, 324
invmodasinh, 325
invmtf, 325
isub, 325
Алфавитный указатель

J
jsonmetadata, 326

L
light_curve, 326
linear_match, 326
link, 327
linstretch, 327
livestack, 327
load, 327
log, 328
ls, 328

M
makepsf, 328
merge, 330
merge_cfa, 330
mirrorx, 330
mirrorx_single, 330
mirrory, 331
modasinh, 331
mtf, 331

N
neg, 332
new, 332
nomad, 332
nozero, 332

O
offset, 333

P
parse, 333
pcc, 333
platesolve, 334
pm, 334
preprocess, 335
preprocess_single, 336
psf, 336

R
register, 337
reloadscripts, 338
requires, 338
resample, 338
rgbcomp, 338
rgradient, 339
rl, 339
rmgreen, 339
rotate, 340
rotatePi, 340

S
satu, 341
save, 341
savebmp, 341
develop, 341
savejpg, 341
savepng, 343
savepnm, 343
savetif, 343
savetif32, 343
savetif8, 343
sb, 343
select, 343
seqapplyreg, 344
seqclean, 345
seqcosme, 345
seqcosme_cfa, 345
seqcrop, 346
seqextract_Green, 346
seqextract_Ha, 346
seqextract_HaOIII, 346
seqfind_cosme, 347
seqfind_cosme_cfa, 348
seqfindstar, 348
seqfixbanding, 348
seqght, 347
seqheader, 347
seqinvght, 349
seqinvmodasinh, 349
seqlinstretch, 349
seqmerge_cfa, 349
seqmodasinh, 350
seqmtf, 350
seqplatesolve, 351
seqpsf, 351
seqrl, 352
seqsb, 352
seqsetmag, 352
seqsplit_cfa, 352
seqstarnet, 353
seqstat, 353
seqsubsky, 353
seqtilt, 354
sequnisetmag, 354
seqvinter, 354
set, 354
set16bits, 355
set32bits, 355
setcompress, 355
setcpu, 355
settext, 356
setfindstar, 356
setmag, 357
setmem, 357
setphot, 358
setref, 358
show, 358
solsys, 359
split, 359
split_cfa, 359
stack, 359
stackall, 361
starnet, 361
start_ls, 362
stat, 362
stop_ls, 362
subsky, 363
synthstar, 363

T
thresh, 364
threshhi, 364
threshlo, 364
tilt, 364

U
unclipstars, 364
unselect, 365
unsetmag, 365
unsharp, 365

V
visu, 365

W
wavelet, 366
wiener, 366
wrecons, 366