Siril

Version 1.4.4

Free-Astro Team

12 juillet 2026

Général

1 Installation **3**
- 1.1 Configuration minimale requise . . . 4
- 1.2 Installation spécifique aux OS . . . 4

2 Interface Graphique Utilisateur **15**
- 2.1 Interface principale . . . 15
- 2.2 Barre d'outils principale . . . 24
- 2.3 Menu Burger . . . 26
- 2.4 Menu Outils . . . 27
- 2.5 Menu traitement d'image . . . 27
- 2.6 Traitement des régions d'intérêt (ROI) . . . 28
- 2.7 Fenêtre d'information sur l'image . . . 30
- 2.8 Raccourcis . . . 30

3 Répertoire de travail **33**

4 Préférences **35**
- 4.1 Préférence (GUI) . . . 35
- 4.2 Préférences (commandes) . . . 60

5 Formats de fichier **69**
- 5.1 Profondeur de bits . . . 69
- 5.2 Format de fichier commun . . . 69
- 5.3 FITS . . . 74
- 5.4 Astro-TIFF . . . 80
- 5.5 SER . . . 83
- 5.6 IRIS PIC . . . 84
- 5.7 PixInsight XISF . . . 84

6 Gestion des couleurs **85**
- 6.1 Pour commencer . . . 85
- 6.2 État de la gestion des couleurs . . . 88
- 6.3 Boîte de dialogue de gestion des couleurs . . . 89
- 6.4 Mode d'évaluation des images ISO 12646 . . . 91
- 6.5 Commandes . . . 92
- 6.6 Matrices de conversion des couleurs . . . 93
- 6.7 Théorie de la gestion des couleurs . . . 95

6.8 Flux de travail . . . 100
6.9 Profils d'espace couleur . . . 102
6.10 L'épreuvage en douceur . . . 106
6.11 Changements dus à l'introduction de la gestion des couleurs . . . 106

7 Séquences **109**
7.1 Un ensemble de deux fichiers FITS ou plus . . . 109
7.2 Un seul fichier SER . . . 110
7.3 Un seul fichier FITS . . . 110
7.4 Charger un séquence . . . 110
7.5 Sélecteur d'image . . . 111
7.6 Exportation de la séquence . . . 113
7.7 Informations sur la séquence . . . 114

8 Définitions et déroulement des opérations **115**

9 Prétraitement **117**
9.1 Conversion . . . 117
9.2 Calibration . . . 122
9.3 Alignement . . . 145
9.4 Drizzle . . . 167
9.5 Empilement . . . 178

10 Traitement **185**
10.1 Étirement de l'image . . . 185
10.2 Étalonnage des couleurs . . . 204
10.3 Autres outils de couleur . . . 226
10.4 Filtres . . . 232
10.5 Traitement des étoiles . . . 276
10.6 Géométrie . . . 286
10.7 Extraction de gradient . . . 291
10.8 Interface GraXpert . . . 296
10.9 Extraction des canaux . . . 302
10.10 Linear Match . . . 308
10.11 Composition RVB . . . 309
10.12 Fusionner les canaux CFA . . . 311
10.13 Pixel Math . . . 314

11 Astrométrie **321**
11.1 Astrométrie . . . 321
11.2 Annotations . . . 337

12 PSF Dynamique **349**
12.1 Candidat étoiles initiale . . . 350
12.2 Modèles . . . 350
12.3 Minimisation . . . 354
12.4 Utiliser . . . 355
12.5 Configuration . . . 356
12.6 Références . . . 358

13 Entête FITS **359**

14 Profil d'intensité **363**
14.1 Profil d'intensité de base . . . 363
14.2 Types de profils . . . 364

14.3 Saisie de coordonnées précises . . . 367
14.4 Mesures . . . 367
14.5 Outil graphique de Siril . . . 368
14.6 Commandes . . . 369

15 Inspecteurs d'images **371**
15.1 Tilt . . . 371
15.2 Inspecteur d'aberration . . . 373

16 Photométrie **377**
16.1 Principes . . . 378
16.2 Photométrie rapide . . . 381
16.3 Courbe de lumière . . . 389

17 Fonction graphique **407**
17.1 Graphique d'alignement . . . 407
17.2 Graphique de photométrie . . . 411
17.3 Options partagées . . . 412
17.4 Interaction avec le graphique . . . 412

18 Statistiques **413**
18.1 Estimateurs . . . 415

19 Siril pour les scientifiques **417**
19.1 Comment Siril peut-il m'aider dans mes recherches ? . . . 417
19.2 Quelle version de Siril dois-je installer ? . . . 417
19.3 Comment puis-je contacter les développeurs ? . . . 417
19.4 Quelle est la précision des résultats de Siril ? . . . 418
19.5 Comment citer Siril ? . . . 418
19.6 Comment puis-je ouvrir des cubes FITS avec Siril ? . . . 418
19.7 Comment mesurer la FWHM dans une image ? . . . 418
19.8 Comment utiliser la détection des étoiles (extraction des sources) ? . . . 419
19.9 Quels sont les outils d'astrométrie disponibles ? . . . 419
19.10 Peut-il traiter les objets du système solaire ? . . . 419
19.11 Siril peut-il fonctionner avec des spectre ? . . . 419
19.12 Siril peut-il remplacer IRAF ? . . . 419

20 Scripts **421**
20.1 Fichiers de Scripts Siril . . . 421
20.2 Scripts Python . . . 429
20.3 Informations essentielles sur les scripts Python à l'intention des utilisateurs . . . 435
20.4 Python Script Key Information for Authors . . . 435
20.5 Editeur de scripts . . . 439
20.6 Exemple de scripts Python . . . 442
20.7 Current list of Python scripts for Siril 1.4 . . . 459
20.8 Obtenir des scripts . . . 463

21 Mode console **465**
21.1 Flux de commande (Pipe) . . . 466

22 Interprétation d'entête **467**
22.1 Exemple de syntaxe . . . 467
22.2 Trouver un fichier avec l'interprétation d'entête . . . 469
22.3 Écriture d'un fichier avec interprétation de l'entête . . . 470
22.4 Utilisation des caractères de remplacement . . . 471

22.5 Gestion des doublons . . . 472

23 Live stacking (empilement en direct) 473
23.1 Livestacking (GUI) . . . 473
23.2 Empilement en direct (Mode console) . . . 475

24 Commandes 477

25 Sirilpy Python Module API 1.0.25 Reference 551
25.1 Dépendances . . . 552
25.2 Guide de codage des Script Siril . . . 552
25.3 Connexion Sirilpy . . . 553
25.4 Modèles de données Sirilpy . . . 574
25.5 Sirilpy Enums . . . 589
25.6 Sirilpy Plotting . . . 594
25.7 Méthodes utilitaires Sirilpy . . . 595
25.8 Assistants GPU Sirilpy . . . 599
25.9 Sirilpy TK GUI . . . 603
25.10 Sirilpy Tkfilebrowser . . . 605
25.11 Exceptions Sirilpy . . . 606

26 Comment signaler des problèmes 609
26.1 Vérifier les listes de modifications et de bugs . . . 609
26.2 Envoyez-nous des informations utiles . . . 609
26.3 Comment nous contacter ? . . . 610

Bibliographie 611

Index des modules Python 613

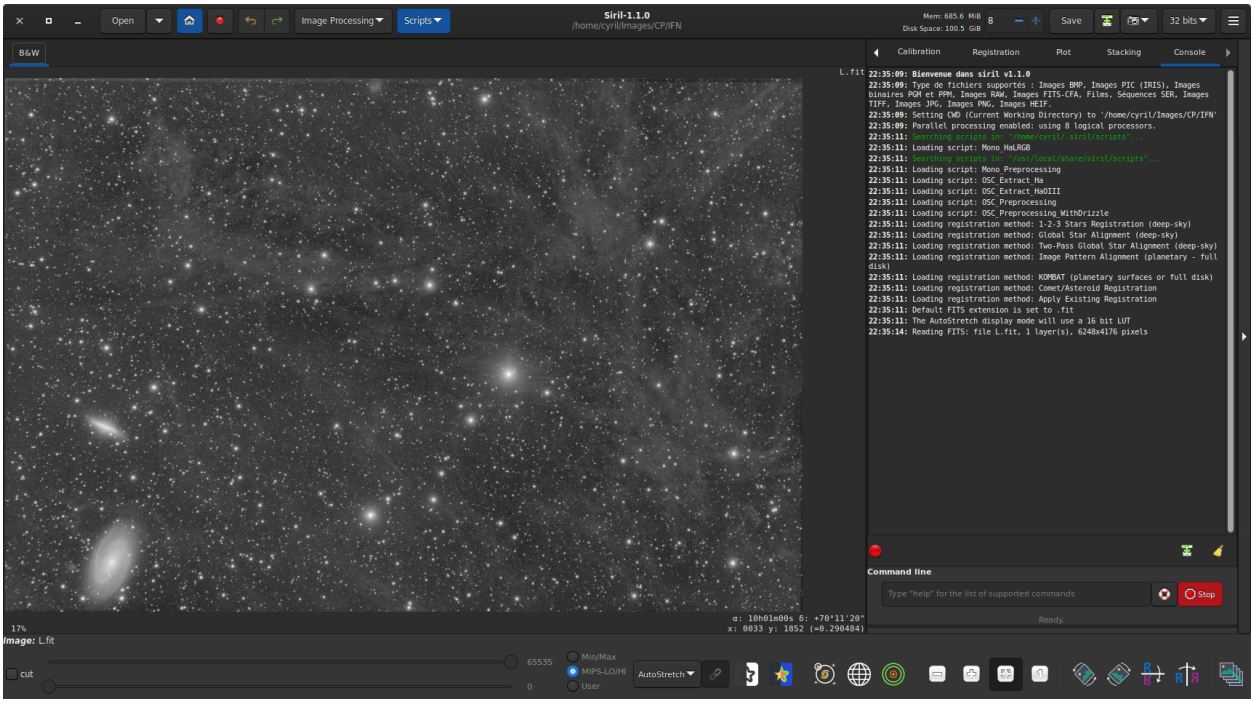
Commands Index 615

Index 619

JOSS 10.21105/joss.07242

Il s'agit de la documentation de la version 1.4.4.

Siril is a deep sky astronomical image processing tool.



Siril can cailbrate, align, stack and enhance pictures from various file formats, even image sequence files (FITS cube, films and SER files). Siril can also analyze images with astrometry and photometry tools.

Le langage de programmation est le C, avec des parties en C++. Le développement principal est effectué avec les versions les plus récentes des bibliothèques partagées sur GNU/Linux. Les contributeurs sont les bienvenus.

Il s'agit de la documentation, qui tente de décrire toutes les fonctions de Siril. Si l'équivalent d'une fonction de l'interface graphique existe sur la ligne de commande, il est donné dans un encart. D'autres ressources utiles peuvent être trouvées sur notre site web principal siril.org.

Astuce

Cette documentation comprend également divers encadrés **conseil** et **avertissement** qui mettent en évidence des informations importantes. En outre, des encadrés **théoriques** ne sont pas indispensables à la compréhension du contenu, mais sont à la disposition des personnes ayant des connaissances en mathématiques et désireuses d'approfondir certains concepts.

Vous pouvez trouvez ici un index des commandes Siril.

Pour signaler tout problème dans la documentation, veuillez ouvrir un ticket à l'adresse suivante : https://gitlab.com/free-astro/siril-doc.

Un problème dans la traduction de la documentation est à signaler ici : https://gitlab.com/free-astro/siril-localized-doc.

CHAPITRE 1

Installation

Chaque version de Siril est distribuée pour les 3 plateformes les plus courantes (Windows, MacOS, GNU / Linux) et peut être téléchargée sur le site de Siril. Mais bien sûr, comme Siril est un logiciel libre vous pouvez compiler l'application à partir des sources.

Astuce

Il peut être utile de vérifier l'intégrité du binaire ou paquet que l'on vient de télécharger. La liste des sommes de contrôle SHA est disponible à cette page, au format json.

À la fin de l'installation, vous pouvez essayer la commande *capabilities* pour en savoir plus sur votre installation.

Ligne de commande Siril

```
capabilities
```

Affiche les fonctionnalités de Siril, en fonction de la compilation et de l'exécution

Comprendre les numéros de version de Siril

À partir de la version 1.0, les versions stables de Siril (telles que 1.0, 1.2, etc.) sont indiquées par des numéros pairs et sont conçues pour une utilisation quotidienne. Les versions de développement, indiquées par des numéros impairs (comme 0.99.0, 1.1.0, etc.), ne sont généralement pas disponibles sous forme de paquets ou d'exécutables binaires, et doivent être compilées par l'utilisateur. Le troisième et dernier chiffre, appelé micro numérotation, correspond au nombre de versions qui ont apporté des corrections de bugs et d'autres petites contributions (telles que 1.0.1, 1.0.2, 1.0.3, etc.).

1.1 Configuration minimale requise

Pour garantir des performances optimales, il est recommandé que votre système réponde aux spécifications minimales suivantes :

- **RAM** : Il est recommandé d'avoir au moins 8 Go de mémoire vive pour tous les systèmes afin d'assurer un bon fonctionnement. Une capacité de mémoire vive plus élevée est bénéfique, car Siril utilise largement la mémoire vive disponible pour les tâches de traitement.
- **Stockage** : Un disque SSD (Solid State Drive) est fortement recommandé pour des vitesses de lecture et d'écriture plus rapides, ce qui se traduit par des performances **un ordre de grandeur** plus rapides que les disques durs traditionnels

Les exigences spécifiques plateforme dépendante sont les suivantes :

- **Windows** : Windows 8.1 ou supérieur.
- **macOS** : macOS 13 Ventura or later.
- **GNU/Linux** : Aucune exigence minimale spécifique n'est indiquée. En général, n'importe quelle distribution moderne devrait suffire.

1.2 Installation spécifique aux OS

Les pages suivantes expliquent la procédure d'installation pour différents OS.

1.2.1 Installation sur GNU/Linux

Installation sur Debian

Le paquet binaire est disponible sur Debian testing et une ancienne version pour stable. Il peut être installé via apt, avec les privilèges du superutilisateur :

Installation sur Ubuntu ou Linux Mint

Dépôts officiels

Pour debian, Siril est disponible dans les dépôts, mais la version peut être obsolète :

```
sudo apt install siril
```

Dépôts PPA

La nouvelle version est donc disponible dans notre PPA, ce qui est la meilleure façon d'installer Siril sur Ubuntu ou Linux Mint :

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

Installation du binaire AppImage

Pour les systèmes GNU/Linux, nous avons décidé de fournir des binaires AppImage (x86_64) et flatpak qui fonctionnent sur les systèmes GNU/Linux. Pour exécuter le binaire AppImage, il suffit de le télécharger et de permettre son exécution à l'aide de la commande :

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

En remplaçant par le chemin correct et x,y et z par les numéros de version. Ensuite, un simple double-clic sur l'AppImage lance Siril.

Installation du flatpak

Another way to install stable version of siril is to use the flatpak, the utility for software deployment and package management for Linux. To install flatpak, type the following command :

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

Ensuite, pour exécuter l'application :

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

1.2.2 Installation sur Microsoft Windows

Installation avec l'installeur

La méthode recommandée pour installer Siril est d'utiliser le programme d'installation fourni qui vous guidera pas à pas.

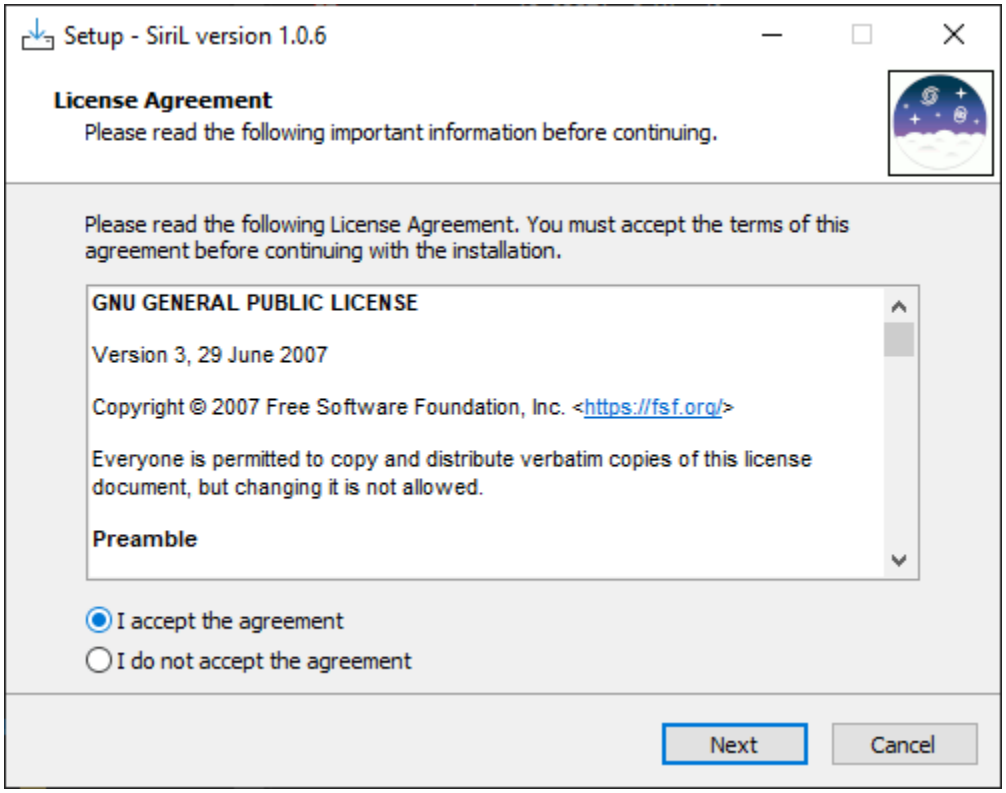


Fig. 1 – Premier écran du programme d'installation, vous devez accepter les termes du contrat de licence pour continuer.

L'assistant d'installation de Siril installera tous les fichiers nécessaires au bon endroit et à la fin vous aurez le choix de créer ou non un raccourci sur le bureau.

> **Note**
>
> Siril sera installé dans C :\Program Files\Siril. Si vous n'avez pas les droits pour installer dans ce dossier, veuillez utiliser la version portable à la place (voir *Installation de la version binaire portable*.)

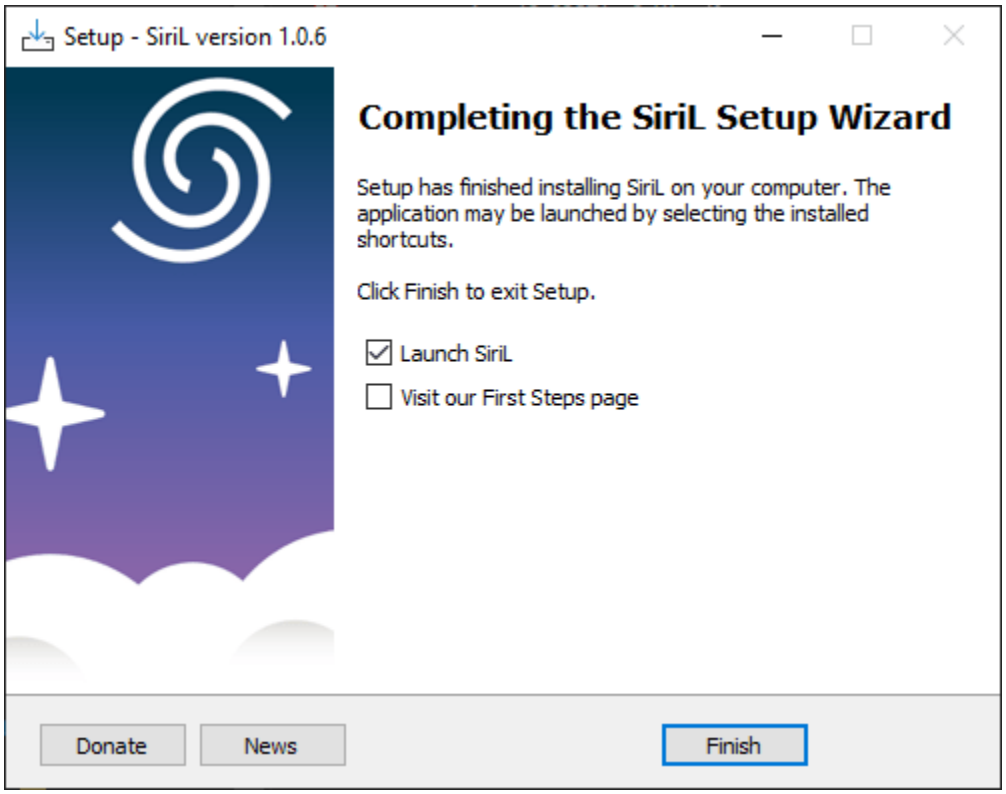


FIG. 2 – Dernier écran de l'installateur. Vous pouvez choisir de lancer Siril immédiatement après l'installation, et d'ouvrir le tutoriel expliquant les premières étapes.

Installation de la version binaire portable

Si vous souhaitez utiliser Siril sans installer toutes sortes de fichiers sur votre ordinateur (par exemple si vous n'avez pas les droits d'administrateur sur la machine), il est recommandé d'utiliser la version portable. Elle se présente sous la forme d'un fichier zip, qu'il vous suffit d'extraire à l'endroit de votre choix, puis d'aller dans le dossier `bin` pour lancer `siril.exe`. Vous pouvez également créer un raccourci sur votre bureau pour faciliter le lancement de l'application.

Avertissement

Attention, en aucun cas vous ne devez déplacer le fichier `exe` , ou tout autre fichier. Sinon Siril ne fonctionnera pas.

Compiler sur Windows avec Msys2

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

Note

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

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

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

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

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

Avertissement

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

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

```
pacman -Syu
```

Installation des dépendances

Pour installer les dépendances, entrez la commande suivante :

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

Note

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

Construire à partir de la source

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

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

Generate the build system and compile the code :

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

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

```
siril
```

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

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

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

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

```
siril
```

Avertissement

Starting with Siril 1.3.6, you will need to have Python installed on your system if you build from sources. The first time you launch a self-built Siril, you must do so from a native Windows environment, not from the MSYS2 shell, in order for Siril to detect the Windows Python installation rather than the MSYS2 one (a notification will appear in the console if this step is skipped, and the sirilpy module will not be initialized). This is a one-time requirement. Once the virtual environment has been set up, you can start Siril from the MSYS2 shell as usual.

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

1.2.3 Installation sur MacOS

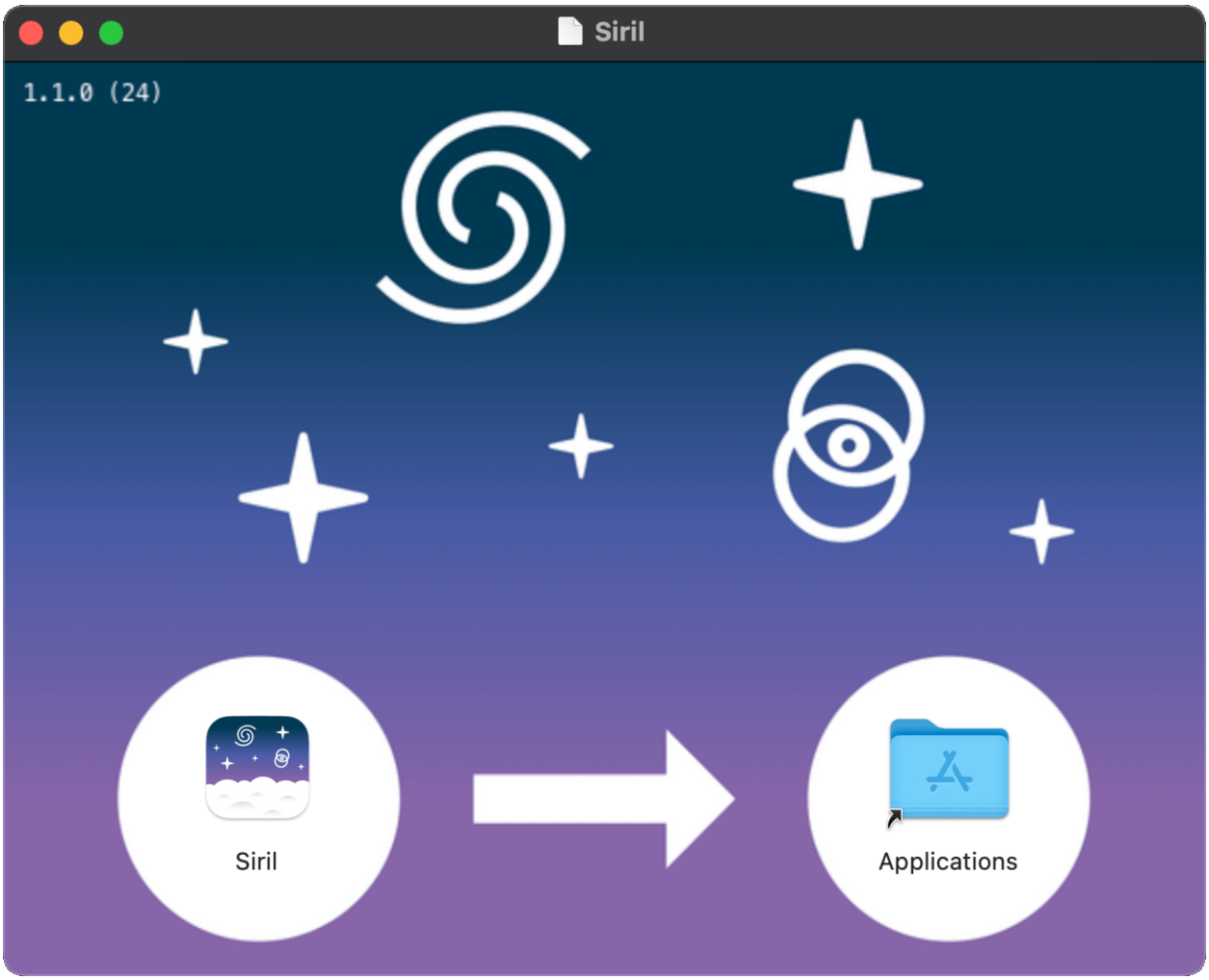
Installation de l'application

L'application macOS est fournie par architecture :

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

Choisissez le lien correspondant à l'architecture de votre processeur et téléchargez l'image disque. Une fois téléchargée, double-cliquez pour l'ouvrir.

Une nouvelle fenêtre s'ouvre. Glissez l'icône `Siril` et déposez-la sur celle des `Applications`.



Félicitation, Siril est maintenant installé.

Installation à partir de Homebrew

Homebrew est similaire à MacPorts et fournit des paquets (aka formulae) à installer, soit en les compilant à partir des sources, soit en utilisant des binaires précompilés (aka bottles). Pour installer Homebrew, cliquez ici. Siril peut être installé avec :

```
brew install siril
```

> **Note**
>
> Veuillez noter qu'il a été annoncé que Homebrew utilise des outils d'analyse. Pour désactiver cela, exécutez : `brew analytics off` Vous pouvez en savoir plus sur Brew Analytics.

1.2.4 Installation depuis les sources

L'installation à partir du code source est nécessaire si vous souhaitez bénéficier des dernières fonctionnalités, si la version précédente est devenue obsolète, si vous souhaitez participer à l'amélioration de Siril ou ne pas utiliser toutes les dépendances.

Récupérer les sources

Les sources sont stockées sur un dépôt git, que vous pouvez télécharger avec cette commande la première fois :

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

Et mettez-le à jour les fois suivantes à l'aide de ces commandes dans le répertoire de base de siril :

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

Dépendances

Siril dépend d'un certain nombre de bibliothèques, dont la plupart devraient être disponibles dans votre distribution Linux ou dans le gestionnaire de paquets de votre choix. Les noms des paquets spécifiques aux systèmes d'exploitation sont indiqués dans chaque section ci-dessous. Les dépendances obligatoires sont :

- gtk+3 (bibliothèque d'interface utilisateur graphique), au moins la version 3.20.
- adwaita-icon-theme (icônes) pour supporter l'apparence de gtk.
- cfitsio (support des images FITS).
- fftw (Bibliothèque de Transformée de Fourier Discrète).
- gsl (The GNU Scientific Library), version 1 ou 2 à partir de la version 0.9.1 ou révision SVN 1040.
- OpenCV et un compilateur C++ pour certaines opérations sur les images.
- Little CMS un système de gestion des couleurs open source
- wcslib pour la gestion du système de coordonnées mondiales, les annotations et l'étalonnage photométrique des couleurs.
- gtksourceview4 for multiline text editing. GtkSourceView adds support for syntax highlighting, undo/redo, file loading and saving, search and replace, a completion system, printing, displaying line numbers, and other features typical of a source code editor.

> **Note**
>
> Même si Siril peut fonctionner en console depuis la version 0.9.9, il est toujours lié aux bibliothèques graphiques, donc vous avez toujours besoin de GTK+ pour le compiler et le faire fonctionner.

Les dépendances optionnelles sont :

- openmp pour le multithreading. Bien qu'optionnelle, cette dépendance est fortement recommandée car les performances seront bien meilleures. Le drapeau de cette option est mis à true par défaut. Cela signifie que si openmp n'est pas installé sur votre machine, vous devez ajouter `-Dopenmp=false` dans la configuration de meson.
- libraw, libtiff, libXISF, libjpeg, libjxl, libpng, libheif for RAW, TIFF, XISF, JPEG, JPEG XL, PNG, HEIF and AVIF images import and export. The libraries are detected at compilation-time.
- FFMS2 pour le support natif des films en tant que séquences d'images. Il permet également d'extraire des images de nombreux types de films, à d'autres fins que l'astronomie. Les versions < 2.20 ont un bug gênant. Il est recommandé d'installer la dernière version.
- ffmpeg (ou libav), fournissant libavformat, libavutil (>= 55.20), libavcodec, libswscale et libswresample pour l'exportation de séquences mp4.
- wcslib pour la gestion du système de coordonnées mondiales, les annotations et l'étalonnage photométrique des couleurs.
- Exiv2 pour gérer les métadonnées des images.
- libcurl pour les requêtes d'astrométrie et de photométrie.
- libgit2 pour maintenir un clone local du dépôt siril-scripts.

Dépendances de la compilation

Pour l'installer à partir du code source, vous devrez installer les paquets de développement de base :

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

Les compilateurs gcc et g++ de cette liste peuvent être remplacés par clang et clang++ (nous les utilisons pour le développement), et probablement d'autres aussi.

Les packages autotools (autoconf, automake, probablement d'autres) peuvent être remplacés par meson.

Processus de compilation général

Siril peut être compilé à l'aide d'autotools ou de meson.

Meson

La méthode recommandée est d'utiliser meson et ninja :

```
meson setup _build --buildtype release

cd _build
ninja
ninja install
```

Pour désactiver certaines dépendances ou fonctionnalités, utilisez les options meson `-Dfeature=false` ou `-Denable-feature=yes` en fonction des cas.

Le tableau ci-dessous énumère toutes les options configurables.

Option	Type	Valeur	Choix	Description
relocatable-bundle	combo	platform-default	['yes', 'no', 'platform-default']	construire avec les ressources considérées comme regroupées sous le même préfixe
openmp	booléen	vrai	N/A	construire avec le support OpenMP
exiv2	booléen	vrai	N/A	construire avec le support exiv2
libgit2	booléen	vrai	N/A	construire avec l'intégration de git libgit2
libraw	booléen	vrai	N/A	construire avec le support LibRaw
libtiff	booléen	vrai	N/A	construire avec le support TIFF
libjpeg	booléen	vrai	N/A	construire avec le support JPEG
libjxl	booléen	vrai	N/A	avec prise en charge de JPEG XL
libpng	booléen	vrai	N/A	construire avec le support PNG
libheif	booléen	vrai	N/A	construire avec le support HEIF
libXISF	booléen	vrai	N/A	construire avec le support XISF
ffms2	booléen	vrai	N/A	construire avec le support FFMS2
ffmpeg	booléen	vrai	N/A	construire avec le support FFmpeg
libcurl	booléen	vrai	N/A	Utiliser libcurl pour l'accès au web
criterion	booléen	faux	N/A	construire avec le support criterion
flatpak-id	chaîne de caractères		N/A	défini par flatpak-builder en cas de compilation d'un flatpak

Autotools

La méthode autotools est bien connue dans le monde unix, une fois que les sources ont été téléchargées et que les prérequis ont été installés, la façon générale de compiler le logiciel est la suivante :

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

éventuellement avec des privilèges de superutilisateur pour la dernière ligne.

Vous pouvez vouloir passer des options spécifiques au compilateur, par exemple si vous voulez que l'optimisation et l'installation se fassent dans `/opt` au lieu du `/usr/local` par défaut :

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

Pour lancer Siril, le nom de la commande est **`siril`** ou **`siril-cli`**.

Installation sur les systèmes de type Debian

Vous pouvez vouloir construire un paquet `.deb` au lieu d'utiliser une version non empaquetée, dans ce cas voir cette help. En particulier, pour installer des dépendances, vous pouvez utiliser la commande :

```
apt build-dep siril
```

Sinon, voici la liste des paquets pour la version actuelle :

— Paquets nécessaires au système de compilation :

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

— Liste des paquets pour les dépendances obligatoires :

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

— Liste des paquets pour les dépendances optionnelles :

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

pour la prise en charge des films (AVI et autres) :

```
libffms2-dev
```

Installation sous Arch Linux

Deux paquets sont disponibles sur AUR : **`siril`** et `siril-git`. Téléchargez le `PKGBUILD` ou le dépôt, installez les dépendances, exécutez makepkg pour construire le paquet et `pacman -U` pour l'installer.

Dépendances (obligatoires et quelques unes optionnelles) :

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

version de LittleCMS

Siril nécessite lcms2 >= 2.14 pour être compilé. Cette version est nécessaire afin d'assurer l'optimisation du code de gestion des couleurs. Une version suffisante est disponible dans les tests Debian et Ubuntu 23.04, mais la version actuelle d'Ubuntu LTS fournit une version plus ancienne, et d'autres distributions de système d'exploitation peuvent également le faire. Si vous vous trouvez sur un système d'exploitation plus ancien qui ne fournit pas lcms2 >= 2.14, vous pouvez soit installer manuellement les packages à partir d'une version plus récente de votre distribution, soit créer et installer lcms2 à partir du code source. La source lcms2 est disponible sur leur dépôt git.

Échecs de compilation

Chaque commit sur le git de Siril est automatiquement construit dans un environnement de construction standard pour Linux, Windows et MacOS en utilisant l'infrastructure gitlab CI. Cela signifie que nous avons une grande confiance dans le fait que la branche master, ainsi que les versions taguées, **se construiront** avec succès dans un environnement de construction correctement configuré avec les dépendances nécessaires installées.

Si vous rencontrez un échec de compilation, il est probable que cela indique un problème avec votre environnement de construction ou des dépendances mal installées - rappelez-vous que de nombreuses distributions nécessitent l'installation séparée de paquets de développement qui contiennent les fichiers d'en-tête nécessaires. Vérifiez le rapport du CI pour le commit git que vous essayez de compiler. Dans le cas improbable où un échec de compilation est affiché, soyez assuré que l'équipe travaille à le corriger. Dans le cas contraire, si le pipeline CI affiche des coches vertes, vous devrez examiner et corriger les problèmes dans votre propre environnement de construction.

Si vous pensez toujours avoir trouvé un problème de compilation qui n'a pas été signalé par le pipeline CI - par exemple si vous construisez sur une plateforme différente, comme BSD, que les développeurs n'utilisent pas régulièrement - alors n'hésitez pas à soulever un problème sur gitlab.

Notez que les problèmes ne doivent être soulevés que pour la branche principale (master) ou les versions taguées. Si vous testez de nouvelles fonctionnalités dans le cadre de demandes de fusion, veuillez fournir un retour d'information dans les commentaires de la demande de fusion concernée.

Système Python

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

CHAPITRE 2

Interface Graphique Utilisateur

L'interface graphique utilisateur (GUI) vous permet de traiter vos images manuellement, ainsi que d'utiliser des scripts ou de taper des commandes. Pour apprendre à utiliser Siril en mode console, veuillez vous référer à cette *section*.

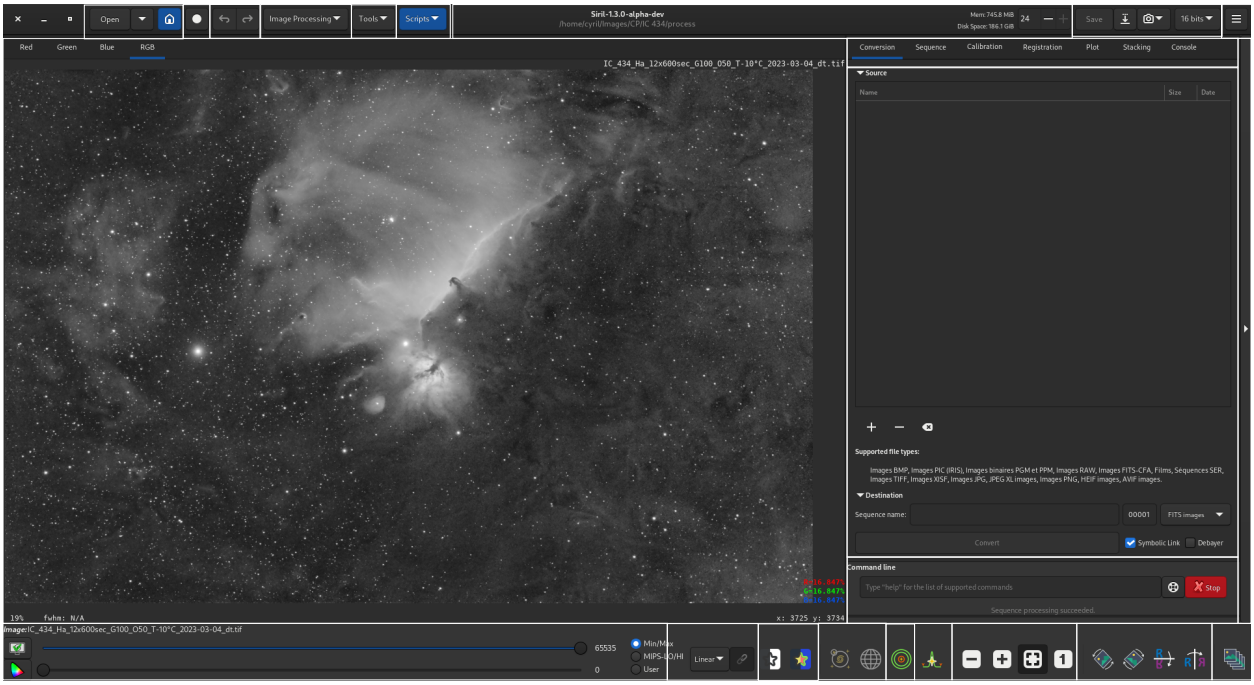
L'interface graphique de Siril est écrite en utilisant GTK, une boîte à outils multiplateforme libre et gratuite pour la création d'interfaces graphiques. Actuellement, la version utilisée est la version 3.

Les sous-sections suivantes vous guideront à travers la fenêtre de l'interface principale et les menus utiles.

2.1 Interface principale

Au lancement de Siril, l'interface principale s'ouvre.

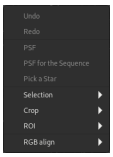
> **Note**
>
> Cliquez n'importe où sur l'image ci-dessous pour afficher ses fonctions.



Zone d'image

Cette zone affiche l'image actuellement chargé. Cliquez sur *Rouge*, *Vert* ou *Bleu* pour passer d'une couche à l'autre (image couleur seulement, un seul onglet *NB* est disponible pour les images mono).

Cliquez avec le bouton droit de la souris sur l'image pour afficher un menu contextuel :



Astuce

Si aucune image n'est chargée, un double-clic sur la zone d'image fait apparaître la boîte de dialogue *Ouvrir*.

2.1.1 Annuler

Annule l'opération la plus récente.

2.1.2 Rétablir

Rétabli l'opération la plus récemment annulée.

2.1.3 PSF

Calcule une PSF pour la sélection actuelle et effectue une photométrie rapide.

2.1.4 PSF de la séquence

Calcule une PSF de la sélection pour toutes les images de la séquence chargée.

2.1.5 Sélectionner une étoile

Applique la détection d'étoiles à la sélection actuelle. Le résultat est ouvert dans la boîte de dialogue*PSF dynamique*. Notez que cette opération est plus souple que la routine de recherche d'étoiles sur l'image complète car elle suppose que vous avez choisi une sélection contenant une étoile : elle peut donc être utilisée pour sélectionner des étoiles qui ont été manquées par la fonction de recherche d'étoiles sur l'image complète.

2.1.6 Sélection

Applique des contraintes à la boîte de sélection.

2.1.7 Recadrer

Fournit une interface aux fonctions *Recadrer* et *Recadrer & rotation*.

Astuce

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

Avertissement

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

2.1.8 ROI

Fournit une interface permettant de définir ou d'effacer la région d'intérêt pour le traitement ROI. Remarque : cette entrée de menu n'est pas disponible si le mode ROI est réglé sur "Automatique à partir de la sélection" dans les préférences.

2.1.9 Aligner RVB

Aligne les canaux RVB de l'image. Cette méthode prend en charge une série de méthodes pour l'alignement :

- **Alignement global des étoiles en 2 passes** Cette méthode utilise la routine d'alignement global des étoiles en 2 passes avec un cadrage COG (par centre de gravité, afin que les dimensions de l'image soient préservées). Cet algorithme s'ajuste avec 8 degrés de liberté, ce qui lui permet de corriger le décalage, la rotation et l'obliquité de l'image. C'est l'algorithme recommandé pour les images contenant des étoiles. Cet algorithme est global et ne nécessite pas de sélection.
- **Alignement KOMBAT** Cette méthode utilise la méthode d'alignement KOMBAT. Elle est principalement destinée à l'alignement planétaire, mais peut également fonctionner pour les images de ciel profond. Elle ne permet qu'un alignement par décalage, mais est rapide. Cet algorithme nécessite une sélection.
- **Alignement sur une étoile** Il s'agit d'un ancien algorithme qui fournit un alignement par décalage uniquement, basé sur le mouvement d'une seule étoile. Il est très rapide, mais l'alignement ne peut pas prendre en compte autant de degrés de liberté que l'alignement global en deux passes. Il peut convenir si vous êtes certain de ne devoir corriger que le décalage de l'image. Cet algorithme nécessite la sélection d'une seule étoile.
- **Alignement par motif de l'image** Cette méthode effectue l'alignement en utilisant les transformées de Fourier de l'image. Elle fonctionne pour les images de ciel profond et les images planétaires, mais ne fournit qu'un alignement de décalage et est assez lente. Cet algorithme utilise une sélection carrée : une sélection doit être effectuée et les sélections non carrées seront corrigées pour être carrées.

Avertissement

Il s'agit d'un outil d'alignement **RVB**. Il n'est pas destiné à aligner des compositions **LRGB** non alignées et ne fonctionnera pas pour cela. Pour ce faire, vous devez soit aligner les calques à composer avant la composition, soit aligner à l'aide de l'outil d'alignement RVB.

Retour à la figure

Ouvrir

Cliquez sur ces icônes (de gauche à droite) pour :

- ouvrir un fichier
- ouvrir un fichier récent
- changer le*répertoire de travail*

Retour à la figure

Livestack

Cliquer sur ce bouton pour démarrer une session *Live stacking (empilement en direct)*.

Retour à la figure

Annuler/Rétablir

Utilisez ces boutons pour annuler/rétablir les dernières actions. Cette fonction n'est disponible que si la dernière action a été effectuée via l'interface graphique, et non en tapant une commande.

Retour à la figure

Traitement de l'image

Cliquez sur ce bouton pour afficher le menu *Traitement*.

Retour à la figure

Outils

Cliquez sur ce bouton pour afficher le menu *Outils*.

Retour à la figure

Scripts

Cliquez sur ce bouton pour afficher et lancer les *scripts*.

Retour à la figure

Barre d'information

Cette barre affiche la version actuelle de Siril et le chemin d'accès au*répertoire de travail* actuel.

— A droite, les information sur la quantité disponible de RAM et d'espace disque sont également données.
— Vous pouvez changer le nombre de fils d'exécution disponible utilisés par Siril en utilisant les signes +/-.

Retour à la figure

Sauvegarder

Ces boutons sont utilisés pour sauvegarder vos résultats :

— sauvegarde (écrase) l'image courante.
— sauvegarde avec un différent nom et/ou extension.
La liste déroulante en bas à droite permet de choisir le type d'image enregistrée. Elle ajoute automatiquement l'extension au nom de fichier. Cependant, en restant en mode *Fichiers d'images pris en charge*, il est possible d'ajouter manuellement n'importe quelle extension prise en charge par Siril et elle sera enregistrée dans le format de fichier correct.
— prendre un instantané de la vue actuelle (telle qu'elle apparaît à l'écran, ce qui signifie que l'étirement de la prévisualisation, le cas échéant, est appliqué). Deux options sont possibles. Soit l'instantané est sauvegardé dans le presse-papiers, soit il est directement copié sur le disque dans le*répertoire de travail*.
— change la profondeur de bit de l'image actuelle. Le choix est entre 16 bits et 32 bits.

Retour à la figure

Menu burger

Ouvre le menu principal, également appelé *menu burger*. Donne accès à *Préférences*, la documentation et bien plus encore.

Retour à la figure

Onglets

Sélectionne un des onglets. Vous pouvez aussi passer d'un onglet à l'autre en utilisant les raccourcis clavier `F1` à `F7`.

Plus de détails peuvent être trouvés ici :

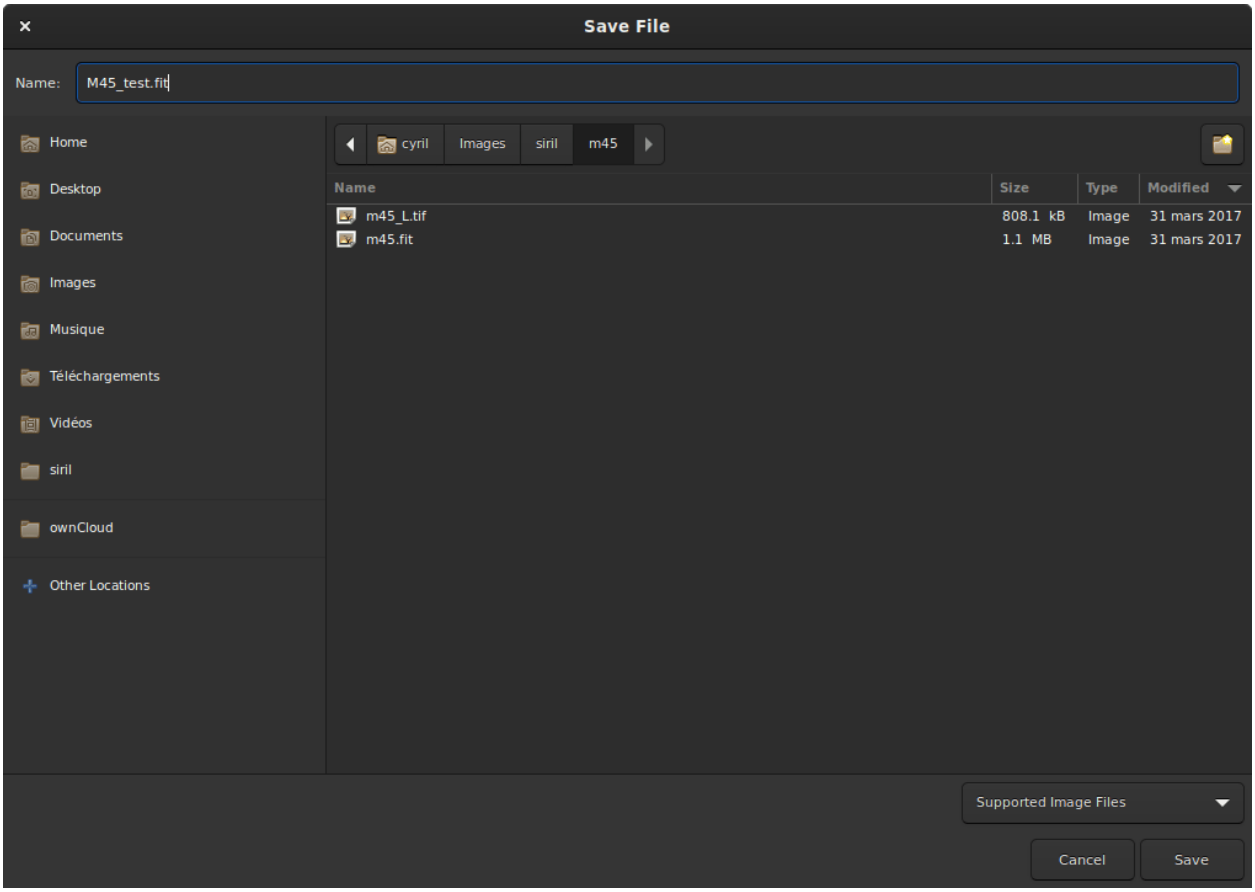


FIG. 1 – Boite de dialogue de sauvegarde.

Onglets	Clés
Conversion	F1
Séquence	F2
Calibration	F3
Alignement	F4
Graphique	F5
Empilement	F6
Console	F7

Retour à la figure

Fenêtre de l'onglet

Affiche les spécificités de l'onglet actuellement sélectionné.

Retour à la figure

Comportement de la souris

Siril utilise les boutons primaire, secondaire et central de la souris. Ces boutons correspondent généralement aux boutons gauche, droit et central (`défilement` de la molette) de la souris physique, mais ils peuvent être affectés différemment dans certaines configurations (par exemple, les gauchers peuvent avoir configuré des boutons gauche et droit inversés).

1. **Bouton primaire de la souris**

   Le bouton primaire de la souris est utilisé à plusieurs fins :
   — Pour sélectionner des zones de l'image
   — En appuyant sur`Ctrl` (ou`Cmd` sur MacOS), pour effectuer un déplacement sur l'image
   — Lorsque l'outil de profilage d'intensité est actif, pour faire glisser la ligne le long de laquelle le profil d'intensité doit être dessiné, et pour sélectionner des points correspondant à des longueurs d'onde / nombres d'onde connues dans le réglage de la spectroscopie
   — Dans l'outil de composition RVB, pour sélectionner le centre de rotation lors de l'utilisation de l'alignement manuel
   — Dans l'outil de suppression des gradients, pour dessiner des échantillons
   — En mode photométrie, pour effectuer une photométrie sur une étoile
   — Pour sélectionner les positions des aperçus de l'alignement manuel
   — Un double clic sur le bouton principal de la souris, alors qu'aucune image n'est chargée, fait apparaître la boîte de dialogue*Ouvrir*.

2. **Bouton secondaire de la souris**

   Le bouton secondaire de la souris est utilisé à plusieurs fins :
   — Pour faire apparaître le menu contextuel de l'image (sauf en mode photométrie)
   — Dans l'outil de suppression de gradient, pour supprimer des échantillons
   — En mode photométrie, pour effectuer une photométrie sur l'étoile sélectionnée pour toutes les images de la séquence actuellement chargée (si aucune séquence n'est chargée, cela n'a pas d'effet)

3. **Bouton central de la souris**

   Le bouton du milieu de la souris est utilisé dans les cas suivants :
   — Pour effectuer un déplacement de l'image au niveau de zoom actuel
   — En appuyant sur `Ctrl` (ou `Cmd` sur MacOS), pour faire une sélection carrée de taille convenable pour la photométrie
   — Un double clic ajuste le niveau de zoom à une valeur prédéfinie. Ce comportement est configurable dans l'onglet*Interface utilisateur* de la boîte de dialogue*Préférences*. Les choix sont :
      — Toujours zoomer pour ajuster
      — Zoom toujours à 100% centré sur le curseur de la souris

— Basculer entre le zoom pour s'adapter et le zoom à 100 % centré sur le curseur de la souris

4. **Molette de défilement**
   — La molette de `défilement` permet de régler le niveau de zoom.

Ligne de commande

Tapez une *commande* et appuyez sur `Entrer`.
— Vous pouvez appuyer sur le bouton au bout de la ligne pour obtenir de l'aide sur l'utilisation.
— Vous pouvez également interrompre le processus en cours d'exécution en cliquant sur le bouton *Arrêter*.
Retour à la figure

Développer

Cliquez sur cette barre pour développer/rétracter toute la zone onglet/fenêtre d'onglet.

Retour à la figure

Curseurs d'image

Utilisez les curseurs supérieur et inférieur pour régler les points blancs et noirs de l'image prévisualisée (en mode linéaire).

Astuce

Cliquez sur le nom de l'*Image* ou de la *Séquence* chargée pour copier son nom dans le presse-papiers (utile pour le coller dans une commande).

Retour à la figure

Mode prévisualisation

Sélectionne le mode d'aperçu de l'image chargée, parmi les choix suivants :
— Linéaire
— Logarithme
— Racine carrée
— Carré
— Asinh
— Auto ajustement (cocher la case Haute définition pour utiliser une profondeur de 24 bits au lieu de celle de 16 bits par défaut)
— Histogramme
En mode Auto ajustement avec des images en couleur, le bouton à bascule situé à droite active/désactive la liaison des canaux. Lorsqu'il est désactivé, les 3 couches sont étirées de manière indépendante afin de donner une image plus équilibrée.

Avertissement

Il s'agit juste d'un aperçu de l'image, et non des données réelles (sauf si le mode linéaire est sélectionné). N'oubliez pas d'étirer vos images avant de les enregistrer.

Retour à la figure

Vues spéciales

Utilisez ces boutons pour afficher les images prévisualisées :
— en couleur inversées
— en fausses couleurs
Retour à la figure

Outils d'astrométrie

Utilisez ces boutons pour afficher :
— astrométrique *annotations*
— grille céleste

> **Avertissement**
>
> L'image chargé doit avoir été résolue astrométriquement pour que ces boutons soient actifs.

Retour à la figure

Photométrie rapide

Utilisez ce bouton pour déclencher le mode *photométrie rapide*.

Retour à la figure

Profil d'intensité

Cette option permet de déclencher le mode *intensity profile*.

Retour à la figure

Zoom

Utilisez ces bouton pour :
— Zoom arrière
— Zoom avant
— Zoom pour s'adapter à l'espace disponible dans la fenêtre
— Zoom à la taille réelle

> **Astuce**
>
> `Ctrl+clic gauche` permet de naviguer dans l'image

> **Astuce**
>
> `Ctrl+défilement de la souris` fera un zoom avant/arrière et `Ctrl` + `0` / `1` fera un zoom pour s'adapter à 100 %.

> **Astuce**
>
> `Ctrl+Maj` et glisser avec le bouton principal de la souris mesurera la distance entre deux points. Si des métadonnées suffisantes sont disponibles, la mesure sera donnée en degrés, minutes et secondes d'arc, sinon elle sera donnée en pixels.

Retour à la figure

Transformations géométriques

Utilisez ces bouton pour :
- Rotation vers la gauche
- Rotation vers la droite
- Miroir autour de l'axe horizontal
- Miroir autour de l'axe vertical

Retour à la figure

Sélecteur d'image

Cliquez sur ce bouton pour ouvrir le *sélecteur d'images*.

Retour à la figure

2.2 Barre d'outils principale

La barre d'outils principale se trouve au bas de la fenêtre principale de Siril. Elle permet d'accéder rapidement aux commandes d'affichage, aux outils d'analyse et aux commandes de base pour la transformation des images.

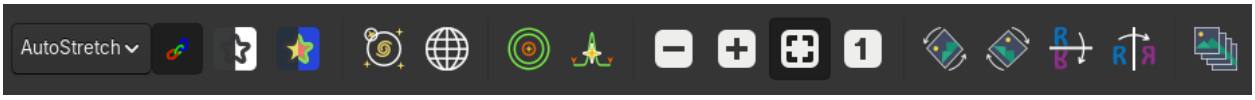


Fig. 2 – Barre d'outils principale de Siril.

Note

La barre d'outils n'est active que lorsqu'une image est chargée en mémoire.

2.2.1 Groupe d'affichage

Mode d'affichage

Le premier contrôle est un bouton de menu déroulant affichant le mode d'affichage actuel (`Linéaire` par défaut). En cliquant dessus, vous ouvrez la liste de tous les modes visuels disponibles dans Siril :
- **Linéaire** — rendu direct des valeurs de pixels. Il s'agit du mode de référence.
- **Logarithme** — applique une fonction de transfert logarithmique, compressant les zones lumineuses et faisant ressortir les détails faibles.
- **Racine carrée** — applique une fonction de transfert racine carrée, une alternative plus douce au logarithme.
- **Carré** — applique une fonction de transfert quadratique, amplifiant les zones claires au détriment des ombres.
- **Asinh** — applique une fonction de transfert sinus hyperbolique inverse, offrant un étirement fluide et naturel.
- **Auto Ajustement** — calcule et applique automatiquement un étirement pour un résultat visuellement agréable. Une sous-option **Haute définition** est disponible pour augmenter la précision du calcul de l'étirement.
- **Histogramme** — applique un étirement basé sur l'histogramme de l'image pour un affichage équilibré.

Avertissement

Seul le mode **Linéaire** (avec les deux curseurs de visualisation réglés sur leurs positions minimale et maximale) reflète les valeurs réelles des pixels. Tous les autres modes sont purement visuels. Étirez l'histogramme avant de l'exporter vers un outil externe.

Liaison de canaux

Immédiatement à droite du sélecteur de mode se trouve un bouton à bascule affichant une icône en forme de maillon de chaîne.

— **Chaîne fermée** — les canaux R, G et B sont liés en mode Auto Ajustement.
— **Chaîne cassée** — les canaux sont étirés indépendamment les uns des autres.

2.2.2 Groupe de rendu visuel

Vue négative

Permet de basculer entre l'affichage normal et l'**affichage négatif** de l'image. Utile pour détecter les artefacts subtils ou les dégradés légers.

Fausse couleur / palette de couleurs arc-en-ciel

Permet de basculer entre l'affichage normal et un **rendu en fausses couleurs** (palette arc-en-ciel). Ce mode permet de visualiser plus clairement les variations d'intensité à travers l'image.

2.2.3 Groupe d'astrométrie

Annotation d'objet

— **Clic gauche** — affiche les noms des objets célestes si les informations WCS (World Coordinate System) sont disponibles dans l'image.
— **Clic droit** — ouvre la liste des catalogues astrométriques disponibles pour sélectionner la source d'annotation.

Grille céleste

Affiche ou masque la **grille de coordonnées célestes** (ascension droite / déclinaison) si les informations WCS sont disponibles dans l'image.

2.2.4 Groupe d'analyse

Photométrie / PSF

Passe en mode *photometry / PSF mode*. Dans ce mode :

— Un clic sur une étoile effectue un ajustement PSF (Point Spread Function) et affiche les paramètres photométriques.
— Si une séquence est chargée, un **clic droit** sur l'image affichée applique l'analyse PSF/photométrie à l'ensemble de la séquence.

Coupe du profil d'intensité

Active l'outil de découpe du profil d'intensité *intensity profile cut tool* : tracez un segment entre deux points de l'image pour afficher un graphique des valeurs de pixels le long de cette ligne.

2.2.5 Groupe Zoom

Zoom arrière

Réduit le facteur de zoom (réduit l'image affichée). Raccourci clavier habituel : -.

Zoom avant

Augmente le facteur de zoom (agrandit l'image affichée). Raccourci clavier habituel : +.

Ajuster à la fenêtre

Lorsque le bouton est activé, l'image est **automatiquement redimensionnée** pour remplir la zone disponible de la fenêtre.

Taille réelle — 1 :1

Affiche l'image à sa **taille réelle** (1 pixel de l'image = 1 pixel de l'écran).

2.2.6 Groupe des transformations géométriques

Tourner de 90° dans le sens inverse des aiguilles d'une montre

Applique une **rotation de 90° dans le sens inverse des aiguilles d'une montre** à l'image.

Tourner de 90° dans le sens des aiguilles d'une montre

Applique une **rotation de 90° dans le sens des aiguilles d'une montre** à l'image.

Miroir horizontal

Applique un **retournement horizontal** (réflexion par rapport à l'axe vertical).

Miroir vertical

Applique un **retournement vertical** (réflexion par rapport à l'axe horizontal).

2.2.7 Groupe de séquences

Liste des images de la séquence

Affiche ou masque le **panneau latéral répertoriant les images de la séquence actuelle**, ainsi que les données d'enregistrement associées à chaque image.

2.3 Menu Burger

Préférences

Ouvre le menu *Préférences*.

Documentation

Ouvre la documentation en ligne.

Vérifier les mises à jour

Vérifier si une version plus récente est disponible.

Raccourcis clavier

Ouvre un panneau rappelant tous les *Raccourcis* disponibles.

À propos de Siril

Ouvre une boîte de dialogue affichant les informations de version et les crédits.

2.4 Menu Outils

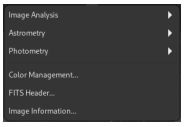


FIG. 3 – Menu Outils

Introduit avec la version 1.4, ce menu regroupe les différents outils de Siril, tels que l'analyse d'images, l'astrométrie, la photométrie, *etc*.

Il se compose de plusieurs menus et sous-menus imbriqués, organisés comme suit :

- **Analyse de l'image**
  - *Statistiques*
  - *Estimation du bruit*
  - *Inspecteur d'aberration*
  - *Montrer le Tilt*
  - *Show Distortions*
  - *PSF Dynamique*
- **Astrométrie**
  - *Résolution astrométrique*
  - *Annotate*
- **Photométrie**
  - *Créer un fichier d'étoiles de comparaison*
  - *Courbe de lumière automatisée*
  - **Créer une sortie**
    - *Courbe de lumière (format ETD)*
    - *Format de fichier AAVSO*
- *Gestions des couleurs*
- *Entête FITS*
- *Information de l'image*

Pour obtenir des informations détaillées sur chaque élément, cliquez sur l'entrée correspondante.

2.5 Menu traitement d'image

Ce menu est expliqué en détail dans la *page suivante*.

2.6 Traitement des régions d'intérêt (ROI)

Siril propose le traitement des régions d'intérêt (ROI) pour certaines fonctions. Le traitement des ROI a deux objectifs :

- Pour les opérations sur les images qui sont lentes à calculer, comme la déconvolution, il permet de ne traiter qu'une petite partie de l'image tout en expérimentant afin de trouver les paramètres idéaux. Cela signifie que l'opération se termine beaucoup plus rapidement sur la petite région d'intérêt et permet de choisir les paramètres souhaités beaucoup plus rapidement.
- Pour les opérations d'image qui affectent l'aspect de l'image, telles que les étirements, les outils de saturation, etc., il peut être pratique d'utiliser le traitement ROI pour comparer côte à côte une zone non traitée et une zone traitée.

Le traitement des ROI n'est pas universel pour toutes les opérations sur les images dans Siril. En fait, il y a beaucoup de cas où cela n'aurait pas de sens, par exemple les opérations géométriques où le contenu de la ROI serait mappé sur une partie différente de l'image. Le traitement des ROI n'est pas supporté pour les opérations d'image qui prennent plusieurs images en entrée, c'est-à-dire *Recomposition des étoiles*, *Composition RVB* ou *Pixel Math*.

2.6.1 Opérations supportant les traitements avec ROI

Les opérations d'image suivantes prennent en charge le traitement des ROI :

- **La déconvolution** La déconvolution peut être très lente, alors le traitement par ROI permet de tester plus rapidement les options sur une petite partie de l'image afin de trouver les meilleurs paramètres.
- **Réduction du bruit** La réduction du bruit peut également être lente à traiter, de sorte que le traitement par ROI permet de tester plus rapidement les options.
- **Étirement asinh** Un utilisateur peut souhaiter examiner l'effet de l'étirement asinh, ou de l'une des opérations d'image suivantes, côte à côte avec une zone non traitée. Le traitement par ROI le permet.
- **Étirement hyperbolique généralisée**
- **Étirement de l'histogramme**
- **Transformation par courbes**
- **Saturation des couleurs**
- **Filtre Médian**
- **Suppression du bruit vert (SCNR)**
- **Filtres préservant les contours**
- **Filtre anti-violet**

Aucun autre processus de traitement ne supporte les ROI.

> **Note**
>
> Le traitement avec ROI n'est généralement pas conçu comme un moyen de traiter une partie seulement d'une image.
>
> Dans les opérations sur les images avec un aperçu mis à jour automatiquement, l'aperçu s'appliquera au xROI, mais lorsque *Appliquer* est cliqué, l'image entière est mise à jour.
>
> Dans les opérations d'image qui n'ont pas de prévisualisation automatique, lorsqu'une ROI est défini, un bouton *Prévisualisation (ROI)* devient disponible. Il s'agit d'un bouton de prévisualisation manuelle, qui vous permet de prévisualiser une série de paramètres dans la ROI jusqu'à ce que vous soyez satisfait. Une fois les paramètres définis, cliquez sur *Appliquer* pour appliquer l'opération à l'ensemble de l'image.

2.6.2 Définition de la ROI

Il existe deux options pour définir la ROI :

- **Manuelle** Cette option exige que la ROI soit défini et effacé manuellement à l'aide du menu du bouton droit de la souris, dans le sous-menu *ROI → Définir la ROI sur la sélection*.
- **Automatique à partir de la sélection** Cette option synchronise automatiquement la ROI avec la sélection en cours et l'efface lorsque la sélection est supprimée.

Ces options sont mutuellement exclusives et l'option préférée peut être configurée dans l'onglet *Interface utilisateur* de la boîte de dialogue *Préférences*.

2.6.3 Effacer la ROI

En mode de réglage manuel de la ROI, l'option permettant d'effacer la ROI se trouve dans le menu du bouton droit de la souris, dans le sous-menu *ROI* → *Effacer ROI*.

En mode de réglage automatique de la zone d'intérêt, il suffit d'effacer la sélection en cours pour que la ROI s'efface.

La ROI s'efface également automatiquement dans les cas suivants :
- Lorsque l'image en cours est fermée,
- Lorsqu'une nouvelle image est ouverte,
- Lorsqu'une nouvelle image est ouverte, lorsqu'une séquence est ouverte et que l'image affichée est modifiée.

2.6.4 Visualisation

Lorsqu'une zone d'intérêt est définie, elle est affichée dans la fenêtre d'image sous la forme d'une boîte rouge en pointillés, dans le même style que la boîte de sélection. Toutefois, lorsqu'une fenêtre de dialogue d'opération d'image est ouverte et que l'opération d'image prend en charge le traitement des ROI (ou si une ROI est défini alors qu'une telle boîte de dialogue est déjà ouverte), le contour de la ROI change de couleur et devient vert, indiquant que le traitement des ROI est activé pour l'opération en cours.

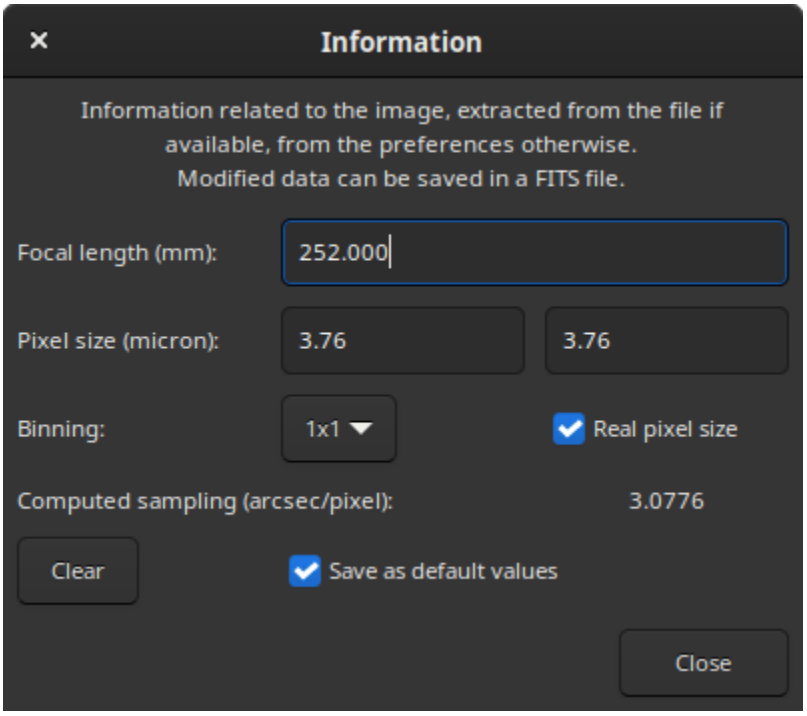
2.6.5 Ligne de commande

Les commandes Siril ignorent la ROI, même si elles sont exécutées en mode GUI à partir de la ligne de commande, et même si la même opération sur l'image prend en charge le traitement de la ROI lorsqu'elle est appelée à partir de l'interface graphique. Le ROI est destiné à donner un aperçu rapide, alors que les commandes sont destinées à opérer sur l'ensemble de l'image. Si une ROI est en place lors de l'exécution d'une commande, elle sera ignorée, mais pas effacée, de sorte qu'il sera toujours disponible pour les opérations ultérieures qui prennent en charge le traitement des ROI.

2.6.6 Transformations de l'espace colorimétrique

Toute opération visant à supprimer, attribuer ou convertir le profil ICC de l'image effacera la ROI, s'il est défini.

2.7 Fenêtre d'information sur l'image



Cette fenêtre fournit des informations sur ce qui est connu sur l'échantillonnage de l'image ouverte. L'échantillonnage, parfois appelé résolution ou échelle, indique combien d'angle du ciel est vu dans un pixel, tel que vu à travers l'instrument. Cela dépend de deux choses : la longueur focale de l'instrument et la taille de pixel du capteur, elle-même dépendant du mode de binning.

Les en-têtes FITS peuvent contenir ces informations si elles ont été donnés au logiciel d'acquisition. Dans ce cas, ce sont les valeurs qui sont affichées dans cette fenêtre. Si ces informations ne sont pas disponibles dans les métadonnées de l'image, car elles étaient inconnues du logiciel d'acquisition, ou simplement parce que le format de fichier ne le prend pas en charge, cette boîte de dialogue sera toujours disponible et remplie de *valeurs par défaut*. Elles peuvent être modifiées et utilisées pour diverses opérations de Siril qui nécessitent des informations d'échantillonnage, par exemple pour afficher la FWHM en secondes d'arc au lieu de pixels.

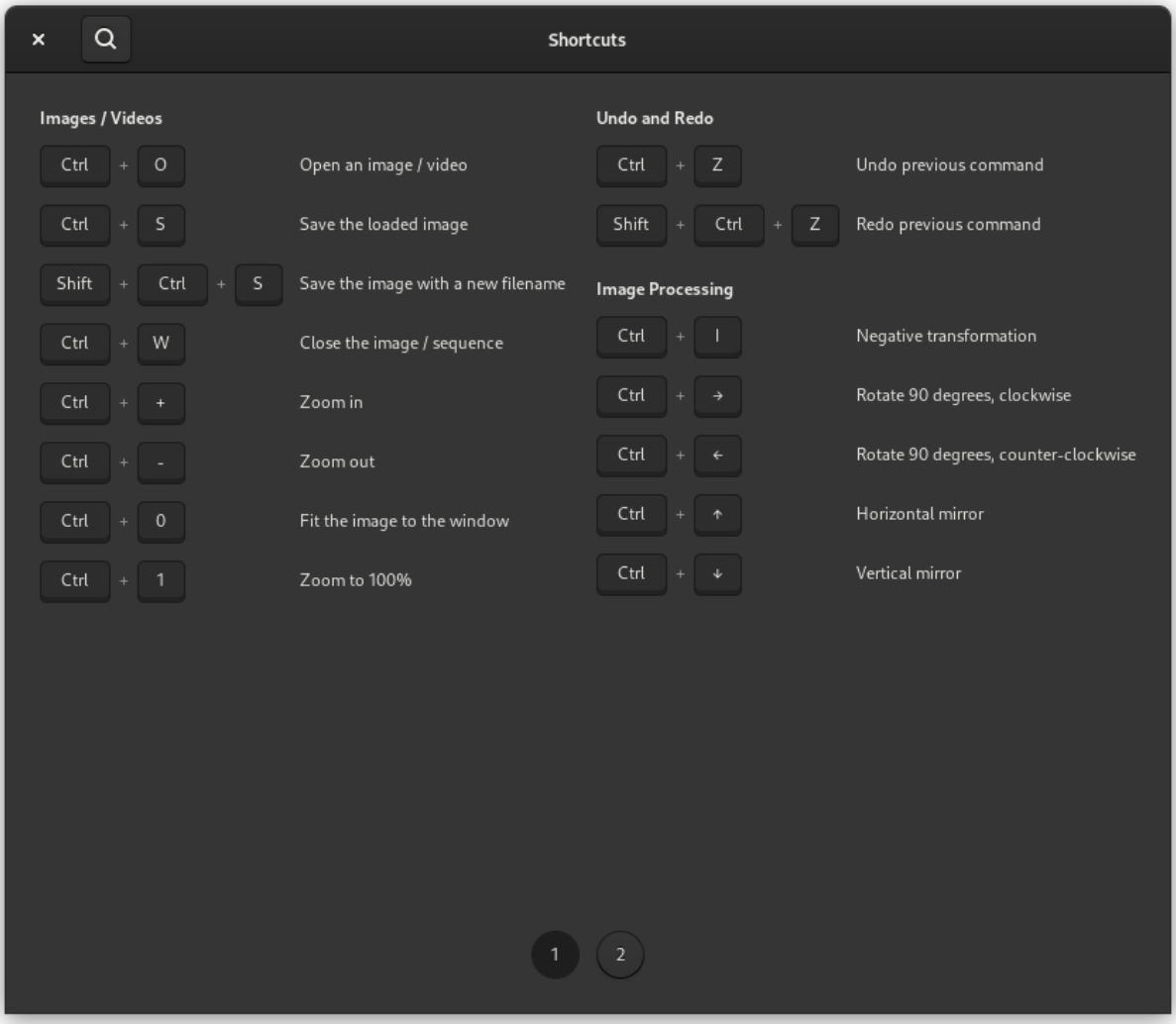
Les valeurs par défaut sont sans binning (1x1), avec une longueur focale et une taille de pixel stockées dans les paramètres. Les valeurs stockées dans les paramètres peuvent être définies à partir de cette boîte de dialogue en activant le bouton *Enregistrer comme valeurs par défaut* avant de cliquer sur *Fermer*. Elles peuvent également être définies en effectuant une résolution astrométrique sur l'image, également appelée *plate solving*, si l'option de mise à jour des valeurs par défaut lorsqu'un résultat est trouvé est activée *dans les préférences*.

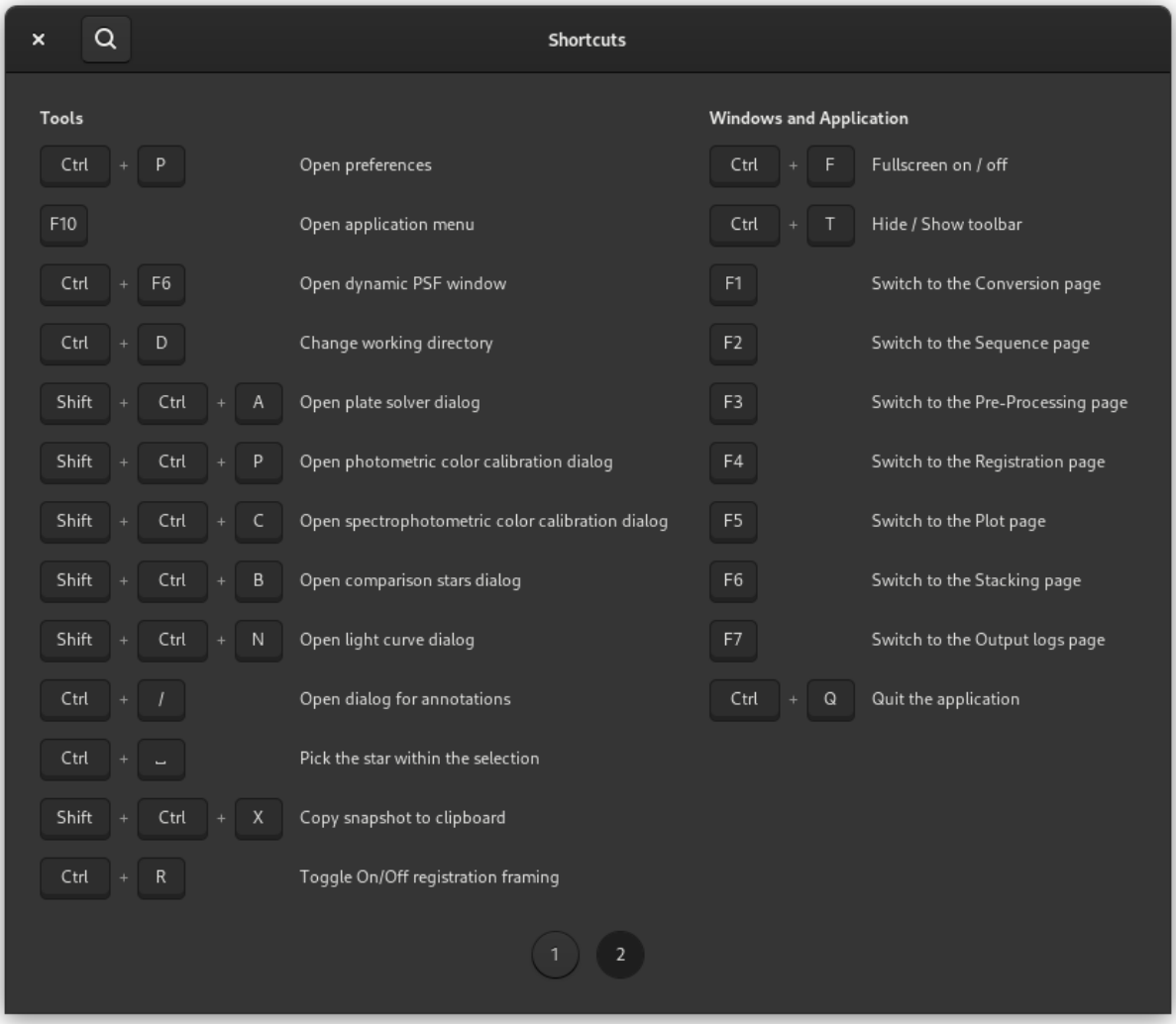
Les valeurs affichées dans cette fenêtre seront stockées dans l'image actuellement chargée et si cette image est enregistrée au format FITS, elles seront stockées dans l'en-tête FITS.

La gestion du binning peut prendre deux formes selon le logiciel d'acquisition : la taille réelle du pixel est donnée mais doit être multipliée par le binning (lorsque *Taille réelle des pixels* est cochée), ou bien la taille du pixel déjà multipliée est donnée (lorsque non cochée).

2.8 Raccourcis

Siril utilise plusieurs raccourcis pour accéder aux outils de traitement ou pour manipuler l'application et/ou les images. Ces raccourcis sont détaillés dans la boite de dialogue *Raccourcis* accessible via la le menu burger ≡.


Shortcuts
Images / Videos
Ctrl + O
Open an image / video
Ctrl + S
Save the loaded image
Shift + Ctrl + S
Save the image with a new filename
Ctrl + W
Close the image / sequence
Ctrl + +
Zoom in
Ctrl + -
Zoom out
Ctrl + 0
Fit the image to the window
Ctrl + 1
Zoom to 100%
Undo and Redo
Ctrl + Z
Undo previous command
Shift + Ctrl + Z
Redo previous command
Image Processing
Ctrl + I
Negative transformation
Ctrl + →
Rotate 90 degrees, clockwise
Ctrl + ←
Rotate 90 degrees, counter-clockwise
Ctrl + ↑
Horizontal mirror
Ctrl + ↓
Vertical mirror
1
2


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

CHAPITRE 3

Répertoire de travail

The Working Directory (`wd`), also known as the Current Working Directory (`cwd`), is the directory in which Siril works. Its choice is a crucial step, especially when using scripts. The wrong choice of `cwd` is responsible for 90% of *script* failures. This folder is selected by clicking on the *Home* button, in the shape of a house : . This is the directory where Siril saves images by default (if no other path is specified) and also the directory where it searches for *sequences*.

Une fois le répertoire sélectionné, son chemin d'accès peut être facilement vérifié dans la barre de titre de la fenêtre de l'application, sous la version utilisée, comme illustré dans la figure ci-dessous.

FIG. 1 – Chemin d'accès au répertoire de travail indiqué dans la barre de titre. Ici, il s'agit d'un chemin sous Linux.

CHAPITRE 4

Préférences

Les préférences sont des paramètres qui sont persistants pour toutes les sessions de Siril, et qui définissent vos choix préférés pour plusieurs des outils.

Depuis la version 1.2.0, elles sont accessibles à la fois depuis l'interface *utilisateur* ou par programme, en utilisant les commandes *set/get*.

Par défaut, le fichier de préférences se trouve à la position suivante :
- `~/.config/siril/configX.Y.ini` (Linux)
- `%LOCALAPPDATA%\siril\configX.Y.ini` (Windows)
- `~/Library/Application Support/org.free-astro.Siril/siril/configX.Y.ini` (MacOS)

Où *X* et *Y* sont les numéros de version majeure et mineure.

Si vous souhaitez avoir plusieurs fichiers de configuration, vous pouvez choisir celui à partir duquel démarrer en ouvrant un terminal et en tapant :

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

Avertissement

Siril doit être dans votre chemin pour utiliser **`siril`** comme dans la ligne ci-dessus. Sinon, utilisez le chemin complet du binaire Siril.

4.1 Préférence (GUI)

Les préférences sont accessibles depuis le menu Burger ☰ ou avec le raccourci `Ctrl + P`. Il y a 10 pages et chaque page représente un thème. Les préférences permettent aux utilisateurs plus ou moins avancés d'optimiser Siril pour qu'il réponde au mieux à leurs besoins. Certains réglages peuvent avoir un impact négatif sur les performances du Siril, il est donc conseillé de ne modifier les réglages que lorsque l'on sait ce que l'on fait. Il y a trois boutons en bas de la boîte de dialogue des préférences : *Réinitialiser* restaure tous les paramètres à leur valeur par défaut, *Annuler* annule les changements en cours et *Appliquer* ferme la boîte de dialogue et enregistre les paramètres.

4.1.1 Dématriçage FITS/SER

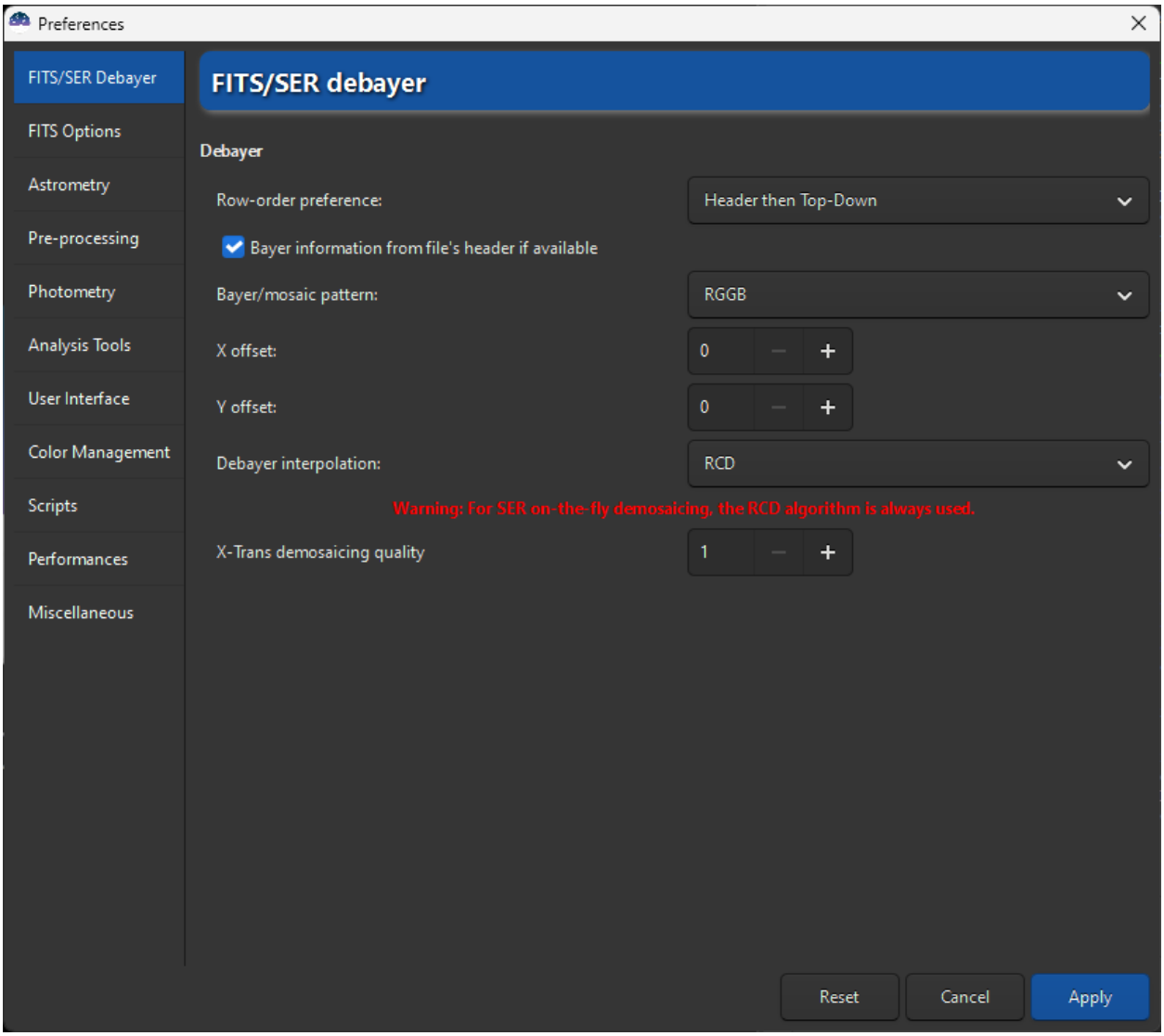


Fig. 1 – Page 1 de la boite de dialogue préférences

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

— **Row-order preference** : You can choose to use header *ROW-ORDER* keywword if present and fallback to either Top-Down or Bottom-Up if not present. Or you can force a preferred orientation. More explanation on this keyword can be found *here*.

— **Bayer/mosaic pattern** : This drop-down menu allows you to choose the type of Bayer matrix used by the camera. It is generally indicated in the manufacturer's information. If *Bayer information from file's header if available* is ticked, the value specified in this dropdown will only be used in case the information is not present in the header. If the option is unticked, it is always used.

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

information.
— **Décalage en Y** : Décalage en Y de la matrice de Bayer.

La modification de ces paramètres entraînera un dématriçage différent à chaque fois. C'est pourquoi il est fortement conseillé de laisser les paramètres par défaut, à moins d'être vraiment sûr de ce que vous faites.

Une autre option qui a moins d'impact sur le résultat final est le choix de l'algorithme de dématriçage proposé dans **Interpolation de dématriçage**. Les choix sont les suivants :

— La **Débayérisation rapide** est l'algorithme le plus rapide disponible dans Siril. Toutefois, les autres algorithmes énumérés ci-dessous sont souvent bien meilleurs.
— **VNG4**, Threshold-Based Variable Number of Gradients (nombre variable de dégradés basé sur un seuil), travaille sur un voisinage de 5x5 pixels autour de chaque pixel source. C'est un très bon algorithme pour les zones uniformes de l'image (comme le fond du ciel), mais il produit des artefacts dans les zones à fort contraste (comme les étoiles).
— **AHD**, Adaptive Homogeneity-Directed, est un autre algorithme de débayérisation bien connu. Cependant, il présente généralement des artefacts en arrière-plan et de mauvaises formes d'étoiles.
— **AMaZE**, Aliasing Minimization and Zipper Elimination, est un algorithme qui donne de bons résultats, en particulier sur les captures à faible bruit.
— **DCB**, Double Corrected Bilinear, un algorithme plus récent, peut présenter des artefacts en arrière-plan comme l'AHD.
— **HPHD**, Heterogeneity-Projection Hard-Decision, est un vieil algorithme qui donne de bons résultats mais qui est assez lent.
— **IGV** et **LMMSE** sont très efficaces lorsqu'ils travaillent avec des images très bruitées. Cependant, IGV a tendance à perdre certaines informations chromatiques, tandis que LMMSE est l'un des algorithme de dématriçage les plus coûteux en termes de calcul et nécessite beaucoup de mémoire.
— **RCD**, Ratio Corrected Demosaicing, vise à atténuer les erreurs de correction des couleurs qui sont courantes dans de nombreuses autres méthodes d'interpolation. Il donne d'excellents résultats pour les bords arrondis, par exemple les étoiles, et c'est donc l'algorithme par défaut utilisé dans Siril.

Pour le capteur X-Trans, un algorithme spécial appelé **Markesteijn** est utilisé quelle que soit la méthode sélectionnée dans les préférences. Pour ce dernier, il est possible de définir la qualité demandée avec l'option **Qualité de dématriçage X-Trans**. Elle définit le nombre de passes pour l'algorithme de dématriçage X-Trans Markesteijn, 1 est la valeur par défaut, 3 peut être légèrement meilleur mais plus lent.

> **Avertissement**
>
> Pour le dématriçage à la volée des fichiers **SER**, l'algorithme RCD est toujours utilisé, quel que soit le choix effectué dans le menu déroulant. Cela permet à Siril d'être plus efficace en termes de vitesse d'exécution et d'offrir une bonne qualité.

4.1.2 Options FITS

La page **Options FITS** regroupe tous les paramètres relatifs au format natif utilisé par Siril.

— **Extension FITS** : Par défaut, la valeur est fixée à **.fit**. Cependant, de nombreux programmes de capture utilisent l'extension **.fits**. Dans ce cas, nous vous conseillons de mettre à jour la valeur. Tous les fichiers créés par Siril auront l'extension définie ici. De plus, seules les séquences ayant l'extension définie dans les préférences peuvent être chargées. Il n'est donc pas possible d'ouvrir une séquence **.fits** et une séquence **.fit** sans mettre à jour cette valeur. Les extensions supportées sont :
  — **.fit**
  — **.fits**
  — **.fts**

  Toutes peuvent être accompagnées de l'extension **.fz** si les fichiers sont compressés.

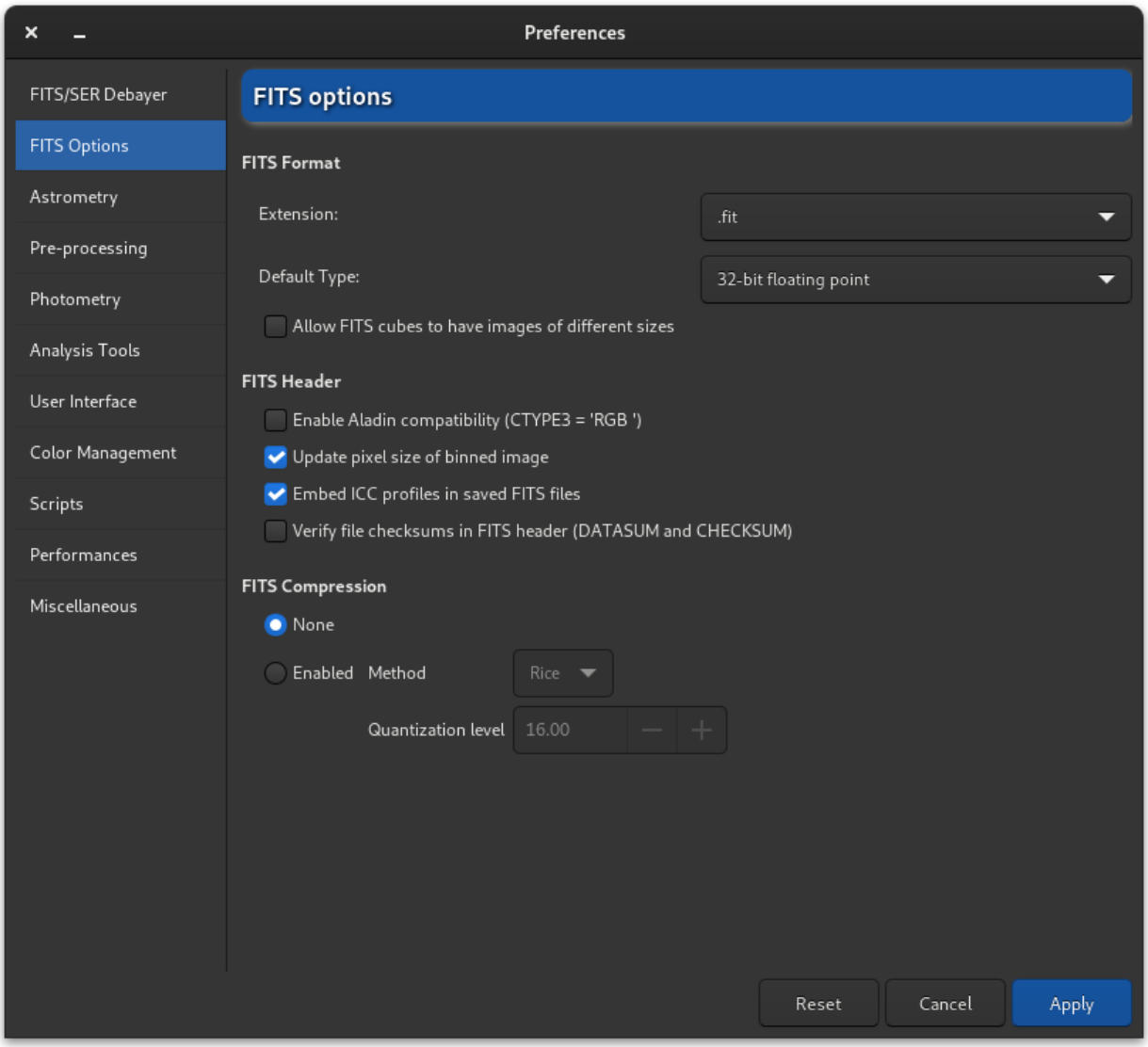


Fig. 2 – Page 2 de la boite de dialogue préférences

Ligne de commande Siril

```
setext extension
```

Définit l'extension utilisée et reconnue par les séquences.

L'argument **extension** peut être "fit", "fts" ou "fits"

— **Type par défaut** : Par défaut, Siril travaille avec des flottants 32 bits dans l'intervalle [0, 1]. C'est la meilleure façon de conserver une grande précision. Cependant, pour des raisons d'espace disque, un utilisateur peut décider de travailler en 16 bits non signés (dans l'intervalle [0, 65535]). Attention cependant, un empilement de 16 bits peut perdre beaucoup d'informations.
— **Autoriser les FITS cubes à avoir des images de tailles différentes** : Cette option peut être utile pour ouvrir des fichiers FITS scientifiques qui n'ont pas été créés par Siril et qui contiennent plusieurs images de dimensions différentes, qui seraient autrement considérés comme des fichiers Siril FITSEQ invalides. Les images du JWST sont un bon exemple de l'utilisation de cette option. Un exemple est donné *ici*.
— **Activer la compatibilité avec Aladin (CTYPE3 = 'RGB ')** : Aladin considère un cube 3D FITS comme une image RVB (composantes Rouge, Bleue et Verte) si le mot-clé FITS `CTYPE3 = 'RGB'` est spécifié dans l'entête. Dans ce cas, toutes les valeurs `BITPIX` sont supportées. Sans le mot-clé FITS `CTYPE3 = 'RGB'`, seuls les cube FITS avec 3 images partageant la même dimension et avec un `BITPIX=8` seront automatiquement détectés comme des FITS RGB.

Avertissement

Cette option peut entrer en conflit avec la fonction d'astrométrie et ne doit être activée que si elle est vraiment nécessaire.

— **Mise à jour de la taille des pixels des images binées** : Utilisée pour le calcul de l'échantillonnage de l'image, la taille du pixel peut être donnée de deux manières différentes : la taille réelle du pixel est donnée mais doit être multipliée par le binning (lorsque la case est cochée), la taille du pixel déjà multipliée est donnée (lorsque la case n'est pas cochée). Cela dépend du logiciel d'acquisition utilisé pour créer le FITS.
— **Embed ICC profiles in saved FITS files** : This should generally be left on, unless you need interoperability with a program that fails to handle FITS with embedded ICC profiles (none are currently known).
— **Verify file checksums in FITS header (DATASUM and CHECKSUM)** : Verify file checksums if they exist. The verification occurs each time the file is read, which may significantly impact Siril's performance when using a sequence.
— **Compression FITS** : La compression peut être intéressante dans certains cas où l'espace disque est un élément clé du traitement. Vous trouverez plus d'informations dans la section dédiée au format FITS, *ici*.
La compression ajoute l'extension `.fz` aux fichiers créés. Siril est capable d'ouvrir une séquence avec l'extension fz sans avoir à changer de valeur dans les préférences.

Ligne de commande Siril

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

Définie si les images sont compressé ou pas.

0 signifie qu'il n'y a pas de compression tandis que **1** active la compression.
Si la compression est activée, le type doit être explicitement écrit avec l'option **-type=** ("rice", "gzip1", "gzip2").

Associée à la compression, la valeur de quantification doit être comprise dans la plage [0, 256].

Par exemple, "setcompress 1 -type=rice 16" définit la compression rice avec une quantification de 16

4.1.3 Astrométrie

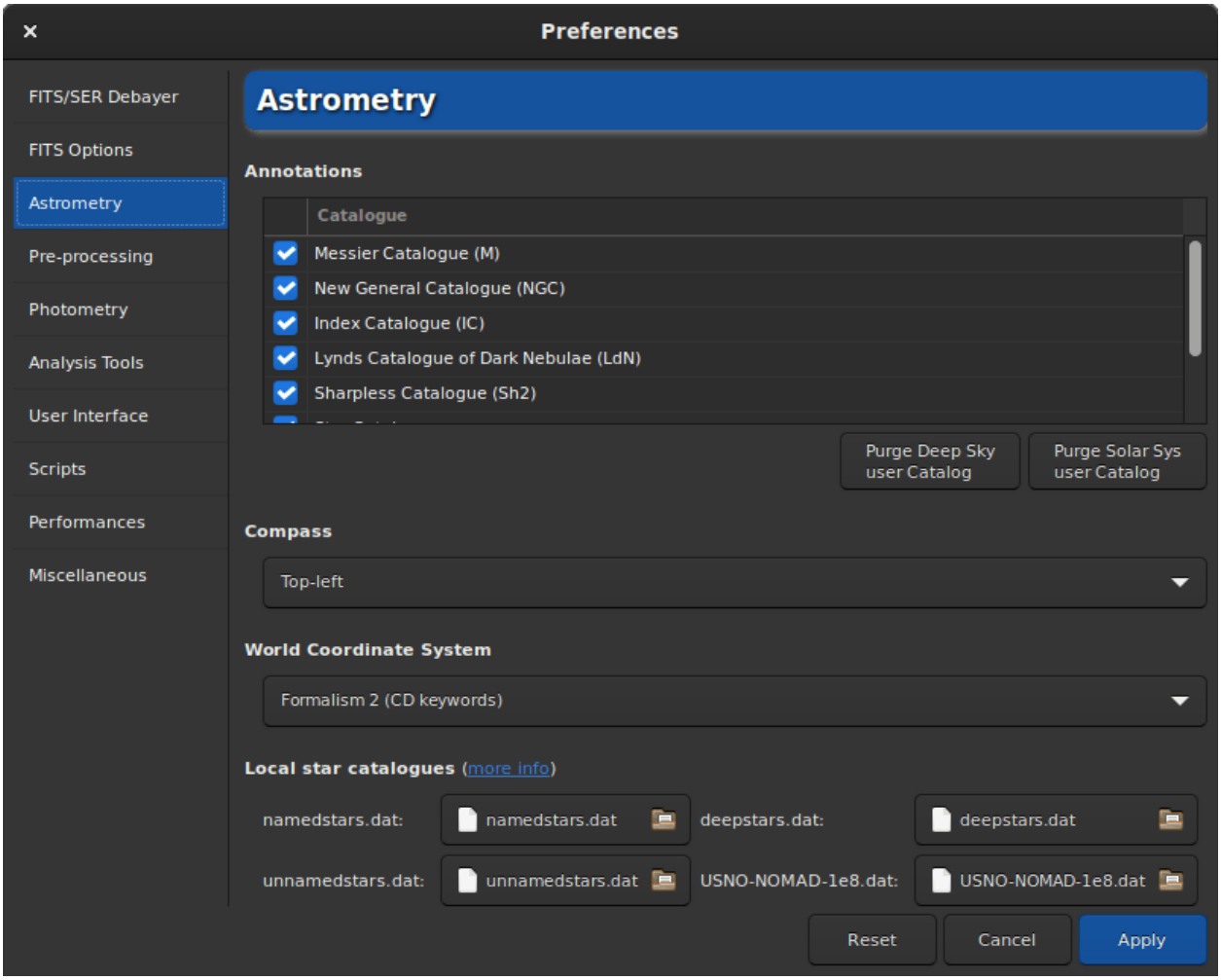


Fig. 3 – Page 3 de la boite de dialogue préférences

Astuce

N'oubliez pas que vous pouvez faire défiler la fenêtre, car il y a des options situées en bas.

This tab contains all the options related to astrometry. Astrometry is a feature strongly implemented in Siril. When the image is solved (i.e. when the astrometry has been successful), it is possible to display the names of the known objects. In particular those listed in the large astronomical catalogs. The annotation part allows to define which catalogs can be used for the display of object names. Currently there are 8 of them, and they can be deselected to be ignored :

— Catalogue de Messier
— New General Catalogue (NGC)

— Catalogue Index (IC)
— Catalogue Lynds des nébuleuses sombres (LdN)
— Catalogue Sharpless
— Catalogue des étoiles les plus brillantes
— Lignes des constellations de l'UAI
— IAU constellations names (positions collected from this page)

En plus de cette liste, il y a deux autres catalogues qui sont remplis par l'utilisateur. L'un concerne les objets du ciel profond, l'autre le système solaire. Ils sont mieux décrits dans la section *annotations* de cette documentation.

Sous la liste des catalogues astronomiques se trouve une option permettant de définir un code d'observatoire UAI préféré. Il s'agit d'un code à 3 caractères qui peut être utilisé pour spécifier un observatoire de l'UAI proche afin d'améliorer la précision de la localisation lors de l'exécution de *conesearch -cat=solsys*. Si l'option *-obscode=* n'est pas fournie, cette valeur par défaut est utilisée de préférence à la valeur géocentrique par défaut qui est utilisée autrement. Le texte de l'option fournit un lien hypertexte vers le site web du Minor Planets Center qui maintient une liste à jour des codes des observatoires de l'UAI, afin que vous puissiez facilement trouver le code le plus proche de votre position.

En cliquant sur le bouton *Afficher les noms d'objets* (seulement si l'image a été résolue astrométriquement), les annotations sont affichées sur l'image. Il est également possible de cliquer sur le bouton qui affiche la grille céleste . Cette dernière ajoute, par défaut, une boussole au centre de l'image. La section **Boussole** permet de définir l'emplacement souhaité pour l'affichage de la boussole.

La section **World Coordinate System** vous permet de choisir
— **Formalisme 1** : Dans le formalisme `PC i_j`, les éléments de la matrice m_{ij} (matrice de transformation linéaire) sont encodés dans des cartes d'en-tête `PC i_j` (à valeurs flottantes), et si comme `CDELT i`. Les indices *i* et *j* sont utilisés sans les zéros initiaux, par exemple `PC 1_1` et `CDELT 1`. Les valeurs par défaut de `PC i_j` sont 1.0 pour $i = j$ et 0.0 sinon. La matrice `PC i_j` ne doit pas être singulière ; elle doit avoir un inverse. De plus, tous les `CDELT i` doivent être non nuls.
— **Formalisme 2** : Les mots-clés `CD i_j` (à valeur flottante) encodent le produit $s_i m_{ij}$. Les indices *i* et *j* sont utilisés sans les zéros initiaux, par exemple `CD 1_1`. La matrice `CD i_j` ne doit pas être singulière ; elle doit avoir un inverse. `CDELT i` et `CROTA i` sont autorisés à coexister avec `CD i_j` pour aider les anciens interprètes de FITS, mais doivent être ignorés par les nouveaux lecteurs.

The **Local star catalogues** part of the dialog window concerns the use of local catalogs to platesolve images. This feature is described in details in the *local star catalogues* section of this documentation.

Dans la section **Astrométrie générale**, un certain nombre d'options sont disponibles pour contrôler à la fois le solveur interne Siril et astrometry.net.

— la première option détermine si la distance focale calculée et la taille du pixel d'entrée sont stockées dans les paramètres comme valeurs par défaut pour les images qui n'ont pas les métadonnées correspondantes, lorsqu'une solution astrométrique est trouvée.
— **Tolérance d'échantillonnage** : pourcentage en dessous et au-dessus de l'échantillonnage prévu à autoriser. L'échantillonnage donné est multiplié ou divisé par 1 + ceci / 100.
— **Degrés de la correction polynomiale** : les solveurs peuvent utiliser une correction polynomiale (SIP) pour travailler avec les aberrations optiques, c'est l'ordre du modèle polynomial. 0 le désactive.
— **Rayon de la cible** : rayon de recherche autorisé autour des coordonnées de la cible pour la résolution (degrés). Non utilisé pour les résolutions à l'aveugle (aucune cible n'est passée).
— **Maximum de secondes pour essayer de résoudre l'image** : temps autorisé pour la résolution de chaque fichier du catalogue. Il ne peut être utilisé comme temps total de résolution seulement si l'astrométrie est configurée pour cela dans le fichier de configuration.

La dernière section est consacrée à la résolution astrométrique par la suite astrometry.net.
— **Ne pas supprimer les tables FITS .xyls** : la liste des étoiles est transmise pour la résolution astrométrique sous la forme d'une table FITS, cochez cette case pour conserver le fichier dans le répertoire de travail.

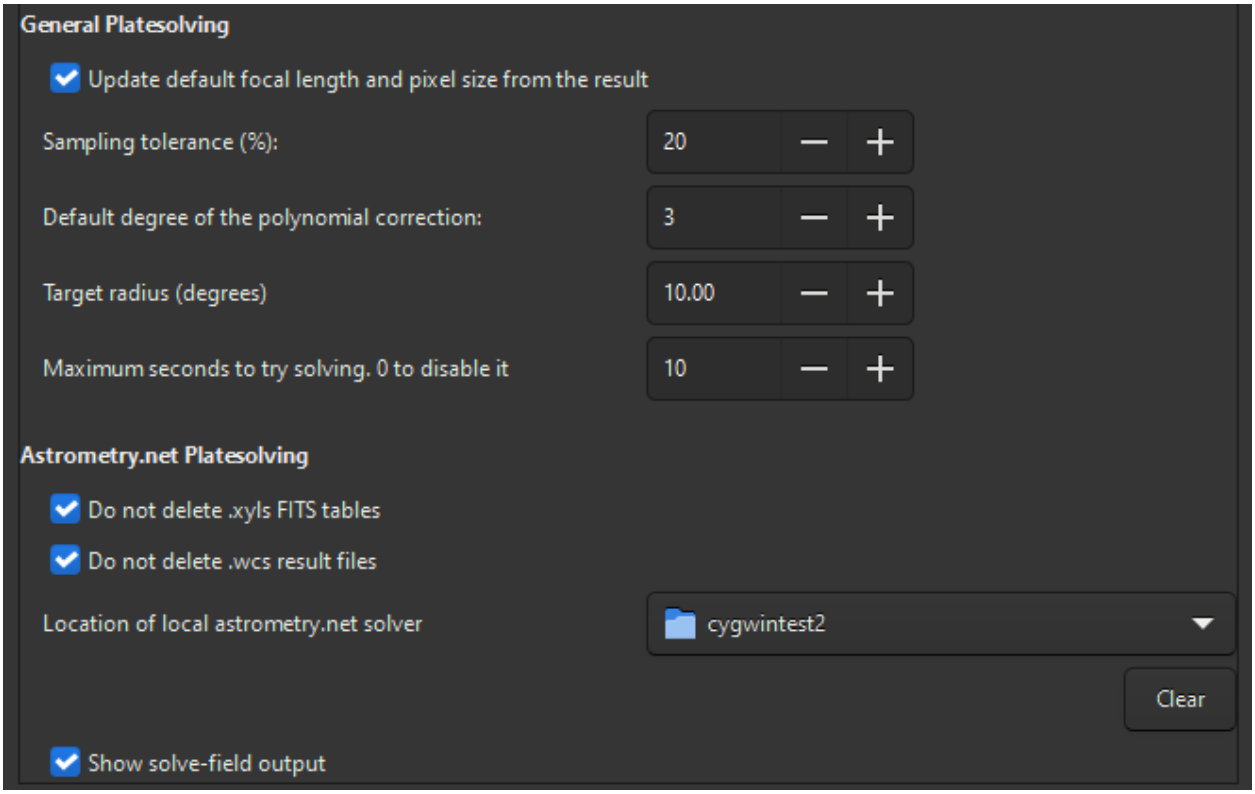


FIG. 4 – Bas de la page 3 de la boîte de dialogue des préférences

— **Ne pas supprimer les fichier de résultat .wcs** : le résultat de la résolution astrométrique est stocké dans une en-tête FITS avec un nom finissant par .wcs. Cocher cette case pour ne pas supprimer ce fichier.
— **Emplacement du solveur local astrometry.net** : Pour utiliser Astrometry.net localement dans Siril, il peut être nécessaire d'indiquer à Siril le chemin où il se trouve. Sur les systèmes UNIX, ce chemin se trouve généralement dans la variable `PATH` et n'est pas nécessaire. Pour Windows, si vous n'avez pas modifié le répertoire d'installation par défaut, c'est-à-dire *%LOCALAPPDATA%\cygwin_ansvr*, Siril le cherchera sans configuration supplémentaire. Si vous avez cygwin et que vous avez compilé astrometry.net à partir des sources, vous devez spécifier l'emplacement de la racine de cygwin ici.
— **Afficher la sortie de la commande solve-field** : imprime la sortie de la résolution astrométrique dans la fenêtre principale de Siril, sinon, seul le résultat sera donné.

4.1.4 Pré-traitement

The pre-processing tab contains all the elements related to the steps that are executed until the stacking. Here it is possible to manage a library of offsets, darks, flats and distortions, the output name of the stacked file or specific corrections for cameras that use the X-Trans sensor.

— **Dark/Bias/Flat/distortion Libraries** : In this section it is possible to load an offset, a dark and a flat that will be used by default in the pre-processing if the button to the right of the text box, *Use it as default* is checked. Each path will also be stored in the reserved keywords `$defbias`, `$defdark`, `$defflat` and `$defdisto` (one token `$`) which can be used when saving a stacking/platesolving result. As far as bias is concerned, it is possible to use more than just a file path. Indeed, in the Siril team we encourage users to use synthetic bias as explained in this tutorial. Several values are then possible as long as the first character entered is the = sign. It is possible to use a fixed integer value like `=500` or a multiplication involving the keyword `$OFFSET` (one token `$`) as long as the latter is actually registered in the FITS file header, like `10*$OFFSET`. More details are given in the tutorial.
— **Empilement par défaut** : Nous définissons ici le nom par défaut que nous voulons donner aux résultats de l'empilement. Il est possible d'utiliser n'importe quelle valeur donnée dans l'en-tête FITS comme mot-clé et de

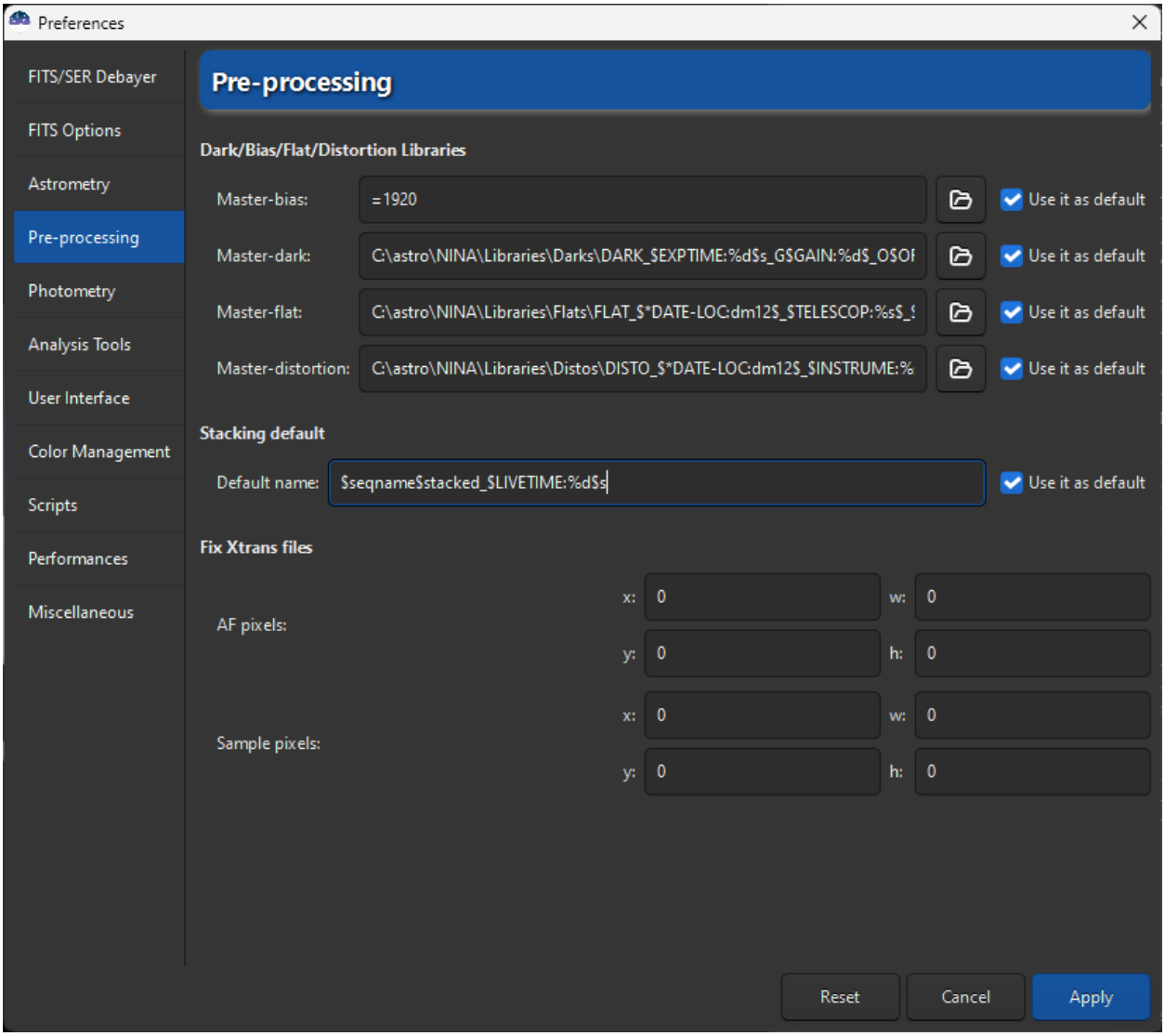


Fig. 5 – Page 4 de la boite de dialogue préférences

l'entourer de jetons $. Si le mot-clé n'existe pas, c'est la variable qui sera utilisée, sinon c'est sa valeur. Un autre mot-clé réservé qui peut être utilisé est `$seqname$`. Il contient le nom de la séquence chargée. Par exemple, le nom par défaut suivant, `$seqname$stacked_$LIVETIME:%d$s` avec un nom de séquence `r_pp_light_` et l'en-tête suivant :

```
...
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)
...
```

produira `r_pp_light_stacked_3900s.fit`.

Note

Both library paths and stacking default support *path parsing*.

— **Corriger les fichiers Xtrans** : Ce champ de réglage est très spécifique et ne concerne que les détenteurs de certains capteurs X-Trans. En effet, certaines images de ces caméras montrent un grand carré au centre des dark et des offsets en raison de la position de la mise au point automatique (AF). Siril dispose d'un algorithme pour l'éliminer pour les caméras suivantes :
- Fujifilm X-T1
- Fujifilm X-T2
- Fujifilm X-T20
- Fujifilm X-Pro2
- Fujifilm X-E3
- Fujifilm X-H1

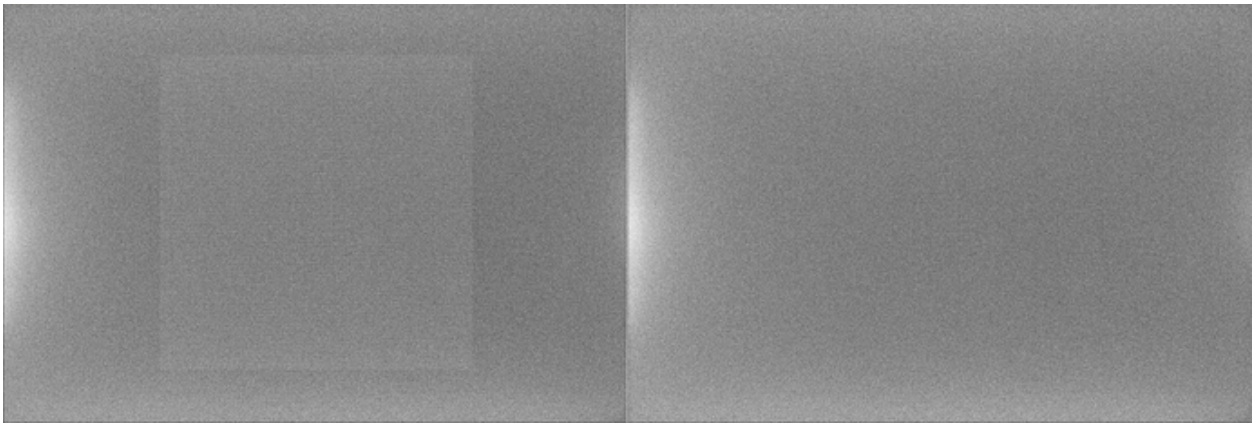


Fig. 6 – Artéfact X-Trans corrigé par l'algorithme de Siril

Dans le cas peu probable où votre appareil photo contient cet artefact et n'est pas pris en charge, il est alors possible de définir la correction à appliquer ici. Le mieux à faire est de contacter l'équipe de développement afin d'obtenir les valeurs à entrer qui correspondraient à votre appareil photo.

4.1.5 Photométrie

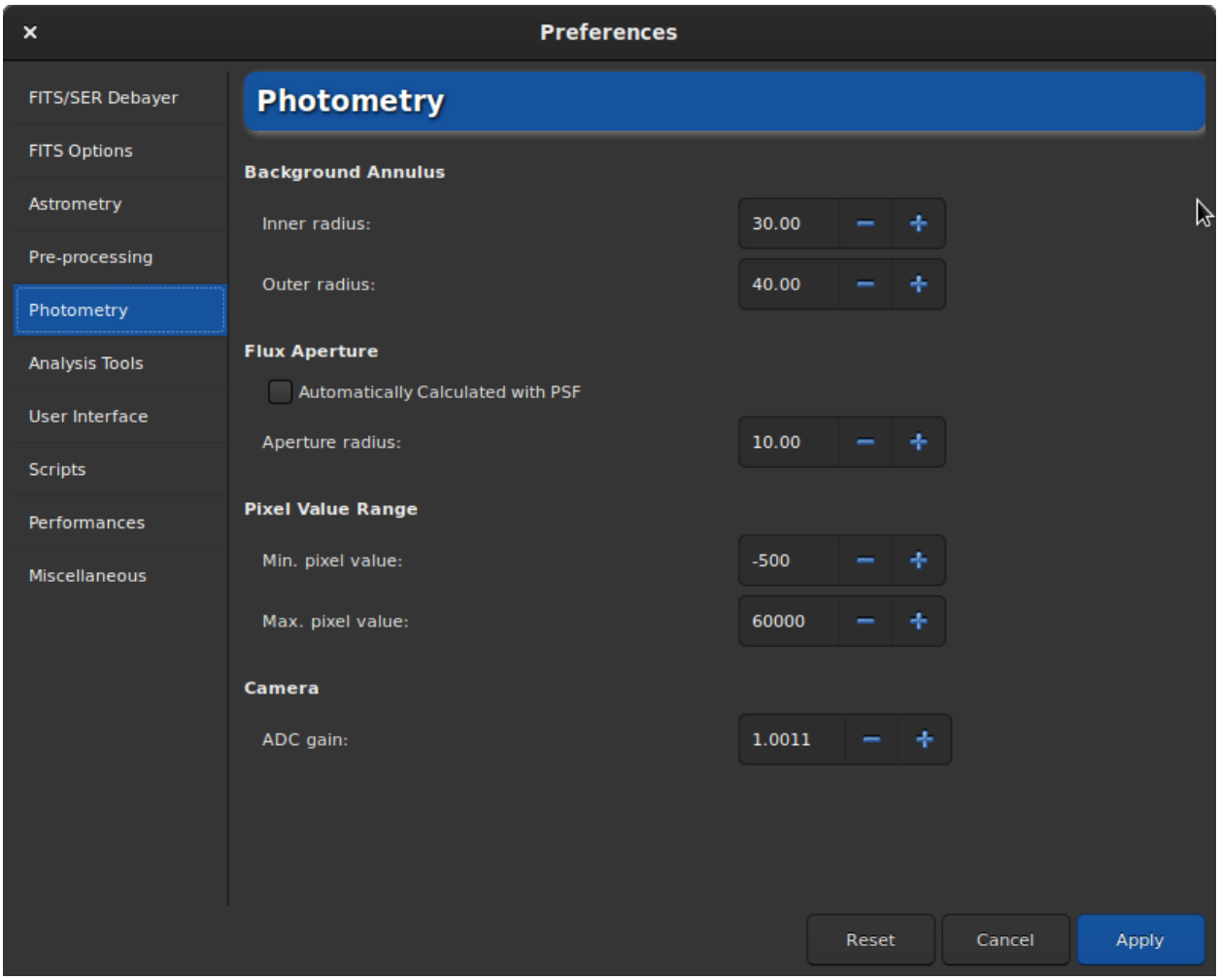


FIG. 7 – Page 5 de la boite de dialogue préférences

La photométrie, qui est l'étude de la lumière, est une autre fonctionnalité très présente dans Siril. Cette section des préférences vous permet de définir les paramètres associés à cet outil.

Le principe de base de la photométrie d'ouverture consiste à additionner le flux observé dans un rayon donné à partir du centre d'un objet, puis à soustraire la contribution totale du fond de ciel dans la même région (calculée dans l'anneau entre le rayon interne et le rayon externe), ne laissant que le flux de l'objet pour calculer une magnitude instrumentale. Cela est décrit plus en détail dans la section *Photométrie* de cette documentation.

— Il est alors possible de modifier le **rayon interne** et le **rayon externe** pour définir une taille qui optimise la valeur du ciel calculée, en essayant d'éviter les étoiles à l'intérieur de l'anneau. Le rayon extérieur **doit** toujours être **plus grand** que le rayon intérieur. Par défaut, le **rayon d'ouverture du flux** est fixé à deux fois la FWHM de la PSF, mais il est possible de désactiver cette fonction et de définir manuellement une valeur fixe.

— **Plage de valeurs des pixels** permet aux utilisateurs de fixer une limite à partir de laquelle le pixel est considéré comme mauvais pour la photométrie. En effet, faire de la photométrie sur des données saturées ne donnera jamais de bons résultats, mais même s'approcher de valeurs élevées peut ne pas être approprié car cela peut être dans le régime non linéaire des capteurs. Une valeur par défaut de 50 000 ADU est fixée pour éviter cette région, mais elle peut varier d'un capteur à l'autre. Les valeurs négatives sont également autorisées, car le bruit peut se rapprocher d'une valeur positive tout en produisant quelques pixels avec des valeurs négatives.

— Enfin, si elle est connue, il est fortement recommandé d'indiquer la valeur du gain du **convertisseur A/N** en

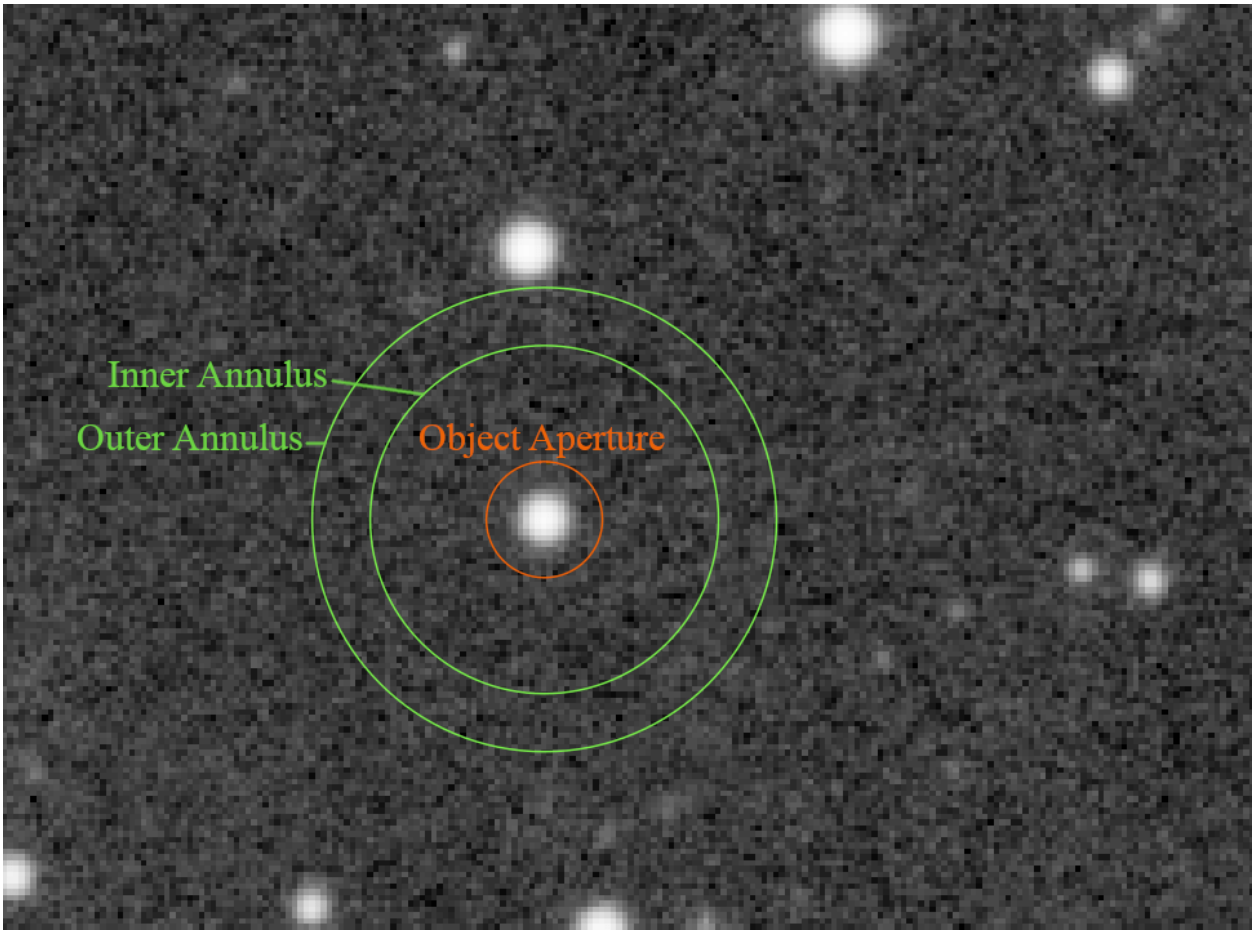


FIG. 8 – Cercle de la photométrie d'ouverture

électrons par ADU : elle est utilisée dans les calculs d'incertitudes, si elle n'est pas déjà fournie dans les en-têtes des images traitées.

4.1.6 Outils d'analyse

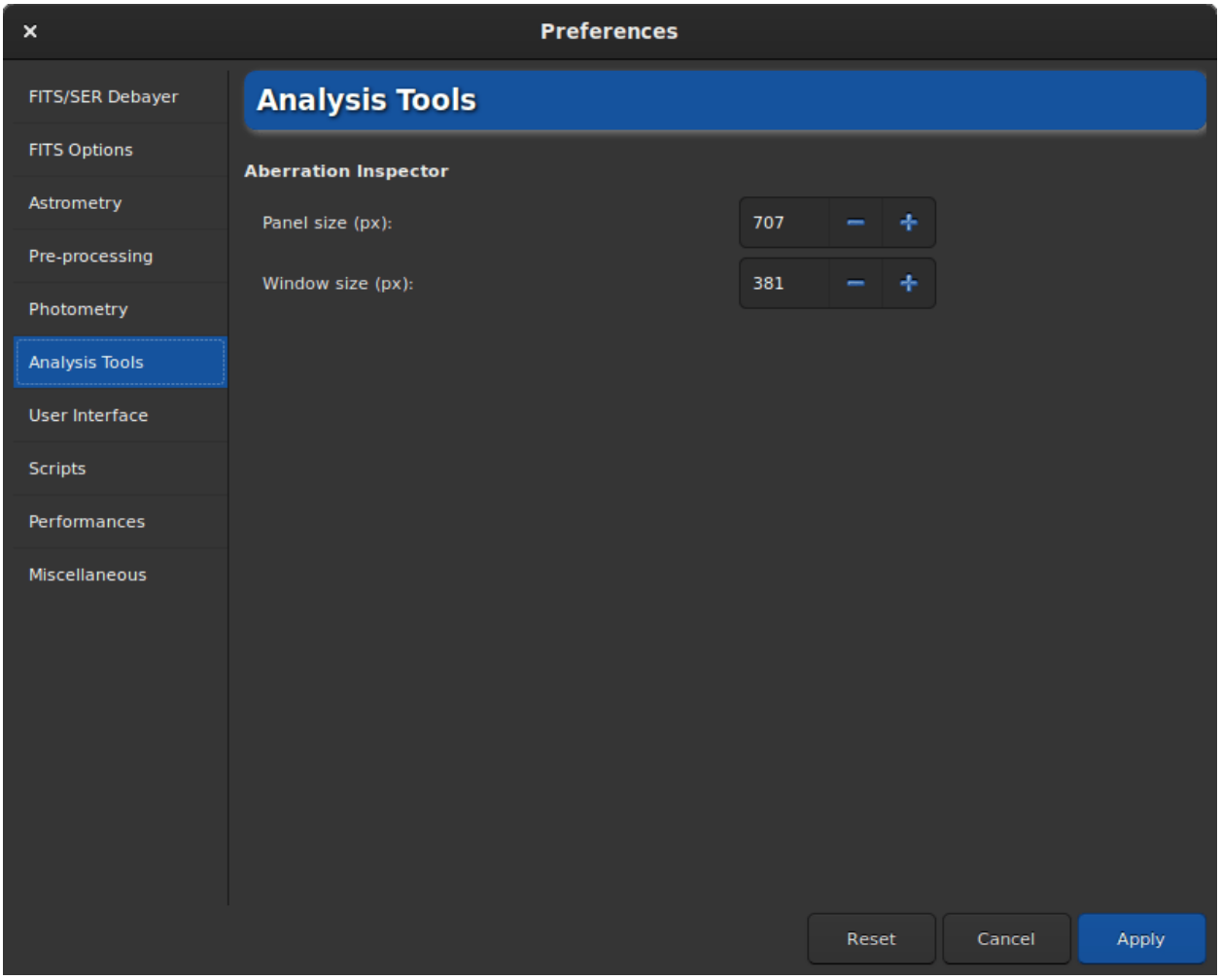


FIG. 9 – Page 6 de la boite de dialogue préférences

Pour le moment, un seul outil d'analyse d'image nécessite des paramètres. Il s'agit de l'outil inspecteur d'aberrations. Dans cet onglet, vous pouvez ajuster :

— La **taille du panneau**, en pixel, qui définit la taille des images qui seront dans les panneaux. Plus la valeur est élevée, plus la taille de l'image dans le panneau est importante. Une valeur trop élevée peut empêcher de voir les défauts des étoiles.

— La **taille de la fenêtre**, également en pixels, définit la taille de la boîte de dialogue. Il est généralement conseillé d'augmenter cette valeur lorsque l'on utilise un écran 4K.

4.1.7 Interface utilisateur

Dans cet onglet sont répertoriés tous les ajustements liés à l'interface utilisateur. Il ne s'agit pas de paramètres ayant un impact sur les processus, mais sur l'aspect et les besoins de l'utilisateur.

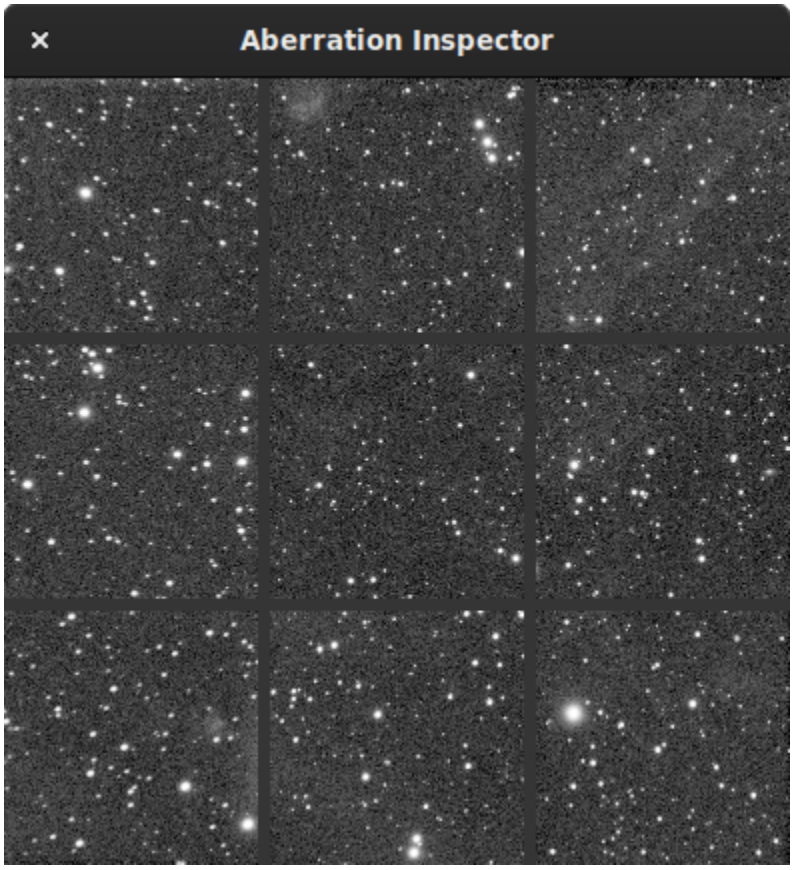


FIG. 10 – Fenêtre de l'inspecteur d'aberration

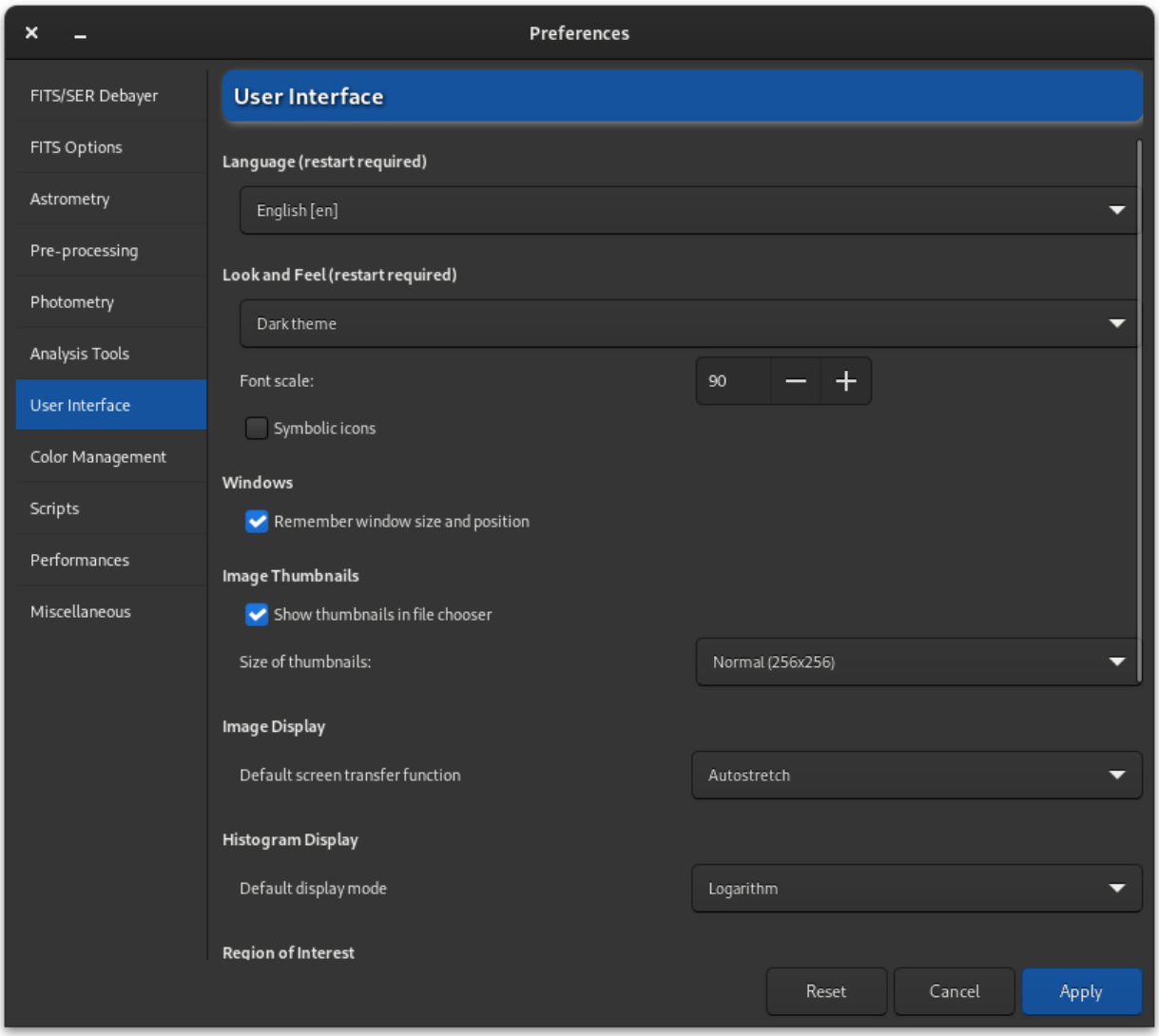


FIG. 11 – Page 7 de la boite de dialogue préférences

Astuce

N'oubliez pas que vous pouvez faire défiler la fenêtre, car il y a des options situées en bas.

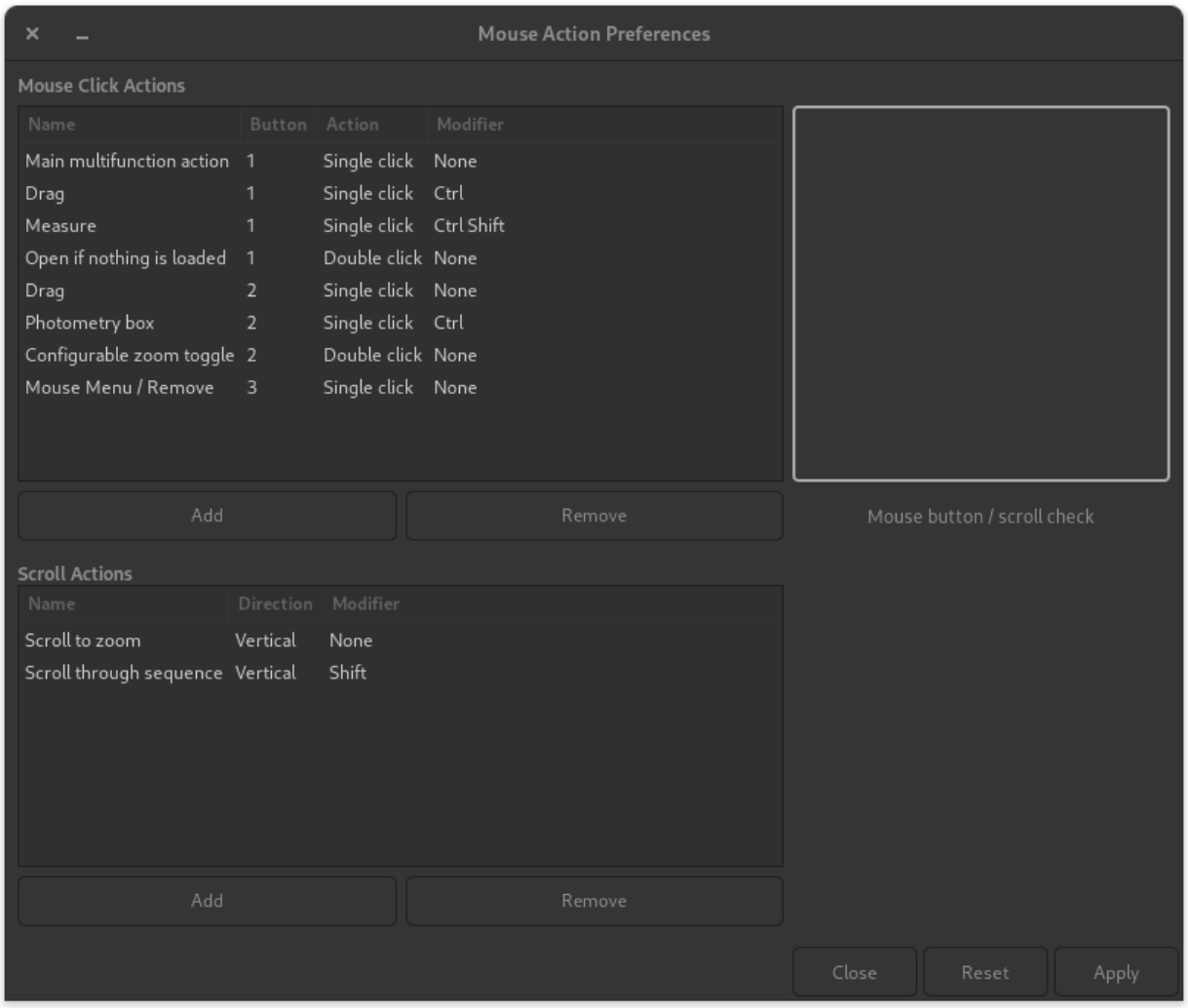
- Par défaut, le **langage** de Siril est définie en fonction de la langue du système. Il est cependant possible de changer la langue et de la définir selon vos besoins, pour autant qu'elle existe. Cependant, gardez à l'esprit que le Siril est développé en anglais.
- Deux thèmes sont disponibles :
  - Le thème sombre (par défaut)
  - Le thème clair

  Le changement de le thème nécessite un redémarrage de l'application pour être complètement opérationnel.
- Il est possible d'ajuster **l'échelle de la police** pour les utilisateurs disposant d'un écran 4K Ultra-HD, ou d'utiliser des **icônes symboliques** pour certaines icônes. Ces réglages nécessitent également un redémarrage de l'application.
- Par défaut, Siril se souvient de la taille et de la position de la fenêtre de l'application à chaque fois que vous la fermez. En cochant le bouton *Se souvenir de la position et la taille des fenêtres* vous pouvez désactiver ce comportement.
- Les miniatures des images sont généralement visibles dans les boîtes de dialogue d'ouverture. Les préférences vous permettent de ne pas les afficher si l'ordinateur a des performances limitées et que l'utilisateur n'en voit pas la nécessité. Vous pouvez également changer la taille de l'affichage des miniatures avec la liste déroulante.
- **La fonction de transfert d'écran par défaut** est le paramètre qui permet d'afficher les images selon les préférences de l'utilisateur. Par défaut, ce paramètre est défini sur *Linéaire*. Comme cela représente réellement l'image, il est recommandé aux débutants de laisser ce paramètre par défaut. Il est facile d'oublier que vous êtes en visualisation auto-ajustée et de ne pas comprendre pourquoi les images sauvegardées ne sont pas telles qu'elles apparaissent à l'écran. Cependant, vous pouvez toujours ajuster la visualisation dans la fenêtre principale.
- Selon le même principe, les histogrammes peuvent être affichés selon deux modes avec **Mode d'affichage par défaut**. Soit le mode *Linéaire*, soit le mode *Logarithme*. Ce dernier peut être très utile avec l'outil d'étirement hyperbolique généralisé. Vous pouvez cependant changer le mode dans chaque fenêtre avec un histogramme. Dans les préférences, il s'agit de définir le comportement par défaut.
- **Région d'intérêt** est le paramètre qui permet de choisir le type de sélection que vous souhaitez appliquer à la ROI : *Manuel* ou *Automatique à partir de la sélection*. Ces options sont décrites en détail dans la section dédiée *Définition de la ROI*.
- **Comportement de la souris** permet de configurer le comportement de la souris. Depuis la version 1.3.0, elle permet de configurer le comportement du double clic du milieu pour définir un niveau de zoom prédéfini. Le niveau de zoom prédéfini peut être réglé soit pour un zoom ajusté, soit pour un zoom à 100 % centré sur le curseur de la souris, soit pour alterner entre un zoom ajusté et un zoom à 100 % centré sur le curseur de la souris. Avec les autres améliorations apportées au contrôle du déplacement sur l'image et du zoom à l'aide du bouton central de la souris, Siril est plus facile à utiliser d'une seule main.
- The **Mouse Configuration** button opens the mouse configuration dialog. This allows you to completely customise your mouse behaviour for Siril by changing default mouse behaviour or adding or removing mouse functions. This can be particularly useful for disabled users to remap existing mouse functions to other buttons or so that they don't need modifier keys, and it can be useful for any user with a mouse with lots of buttons (or two-way scroll wheels) to assign new actions (such as save, undo, scroll through the active sequence) to mouse buttons or scroll axes in a way that has not previously been possible, or with two-way scroll wheels.

  You need to click through a reminder that this tool can remove or change mouse behaviour so that your settings no longer match the documentation. The tool has a certain amount of sanity checking built in (it will reject duplicate assignments, it will insist that the main mouse action used for interacting with the image is included, and it will check for clashes between single click and double click actions assigned to the same button and modifier. If you really mess up your configuration there is a reset button that will reset all mouse behaviour to the defaults.

  Mouse button numbering is not always obvious, especially with mice with many buttons, as the scroll wheel directions are accounted for in button numbering, so a scratchpad area is provided to the right of the dialog. If

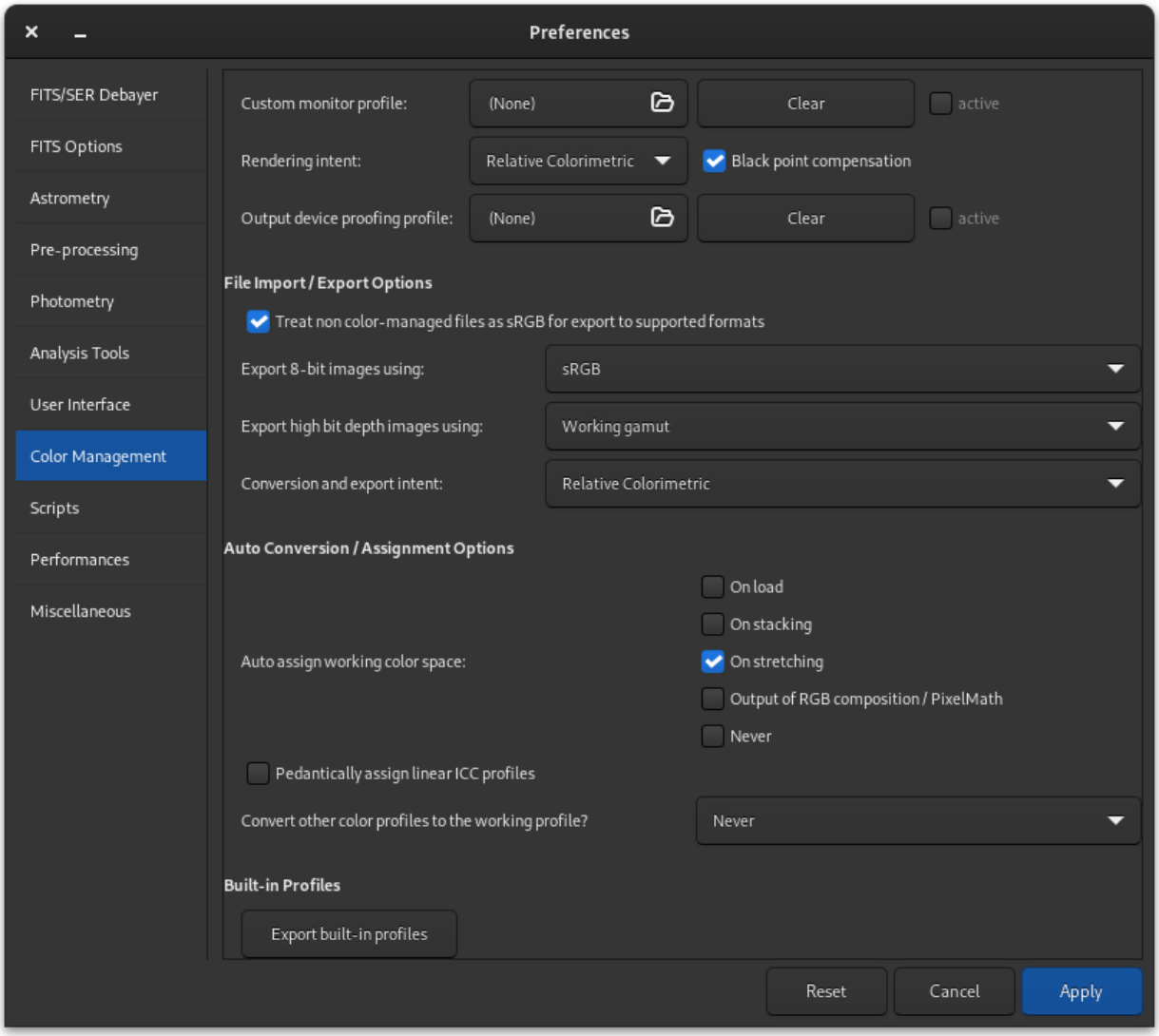
you click any mouse button here, its number will be shown in the label. This is the same number you need to use when assigning functions to that mouse button.



> **Astuce**
>
> Note that this configuration applies only to mouse events on the main drawing area (where the image is displayed). Regardless of any mouse actions configured here, the mouse behaviour will still be the same elsewhere in the application. So for example if you want to universally swap left and right mouse buttons, you can't do that here, but you may be able to do so in your Operating System settings or mouse driver config.

— Vous pouvez configurer certaines couleurs de textes et de dessins à afficher sur l'image dans la section **Couleurs**. Pour ce faire, il suffit de cliquer sur le bouton de couleur et de choisir la couleur de votre choix. Ce choix porte sur 5 éléments :
  — Échantillons de l'extraction de gradient
  — Annotations standards
  — Annotations des objets du ciel profond
  — Annotation des objets du système solaire
  — Annotation des objets temporaires

4.1.8 Gestion des couleurs



Espaces de travail des couleurs

Espace colorimétrique préféré

Cette option sélectionne l'ensemble des espaces couleur de travail, à savoir un espace de travail RVB linéaire, un espace de travail RVB non linéaire avec la courbe de réponse aux tons (CRT) standard de l'espace couleur, et un espace de travail gris avec une CRT adaptée à l'espace de travail RVB. Les options sont sRGB, qui convient le mieux pour travailler avec des images destinées au WWW ; Rec2020, qui est l'espace colorimétrique à large gamut recommandé par Siril ; et Custom, qui vous permet de définir votre propre ensemble de profils à l'aide des sélecteurs de fichiers.

Profils de couleur externes

Ces sélecteurs de fichiers ne sont utiles qu'en combinaison avec l'option "À partir de fichiers" sélectionnée ci-dessus pour votre espace colorimétrique préféré.

— **Profil ICC RVB** : Ce sélecteur de fichiers vous permet de choisir le profil ICC RVB correspondant à votre espace colorimétrique préféré.

— **Profil ICC de gris** : Ce sélecteur de fichiers vous permet de choisir un profil ICC de gris. Ce profil doit avoir une courbe de réponse des tons (CRT) correspondant à celle de l'espace colorimétrique RVB que vous avez choisi.

Un exemple de définition d'un gamut RVB personnalisée correspondant à Pro Photo RGB est disponible sur la page *des profils de couleur*.

Visualisation Gamut

Ce bouton ouvre une image de référence comparant les gamuts offerts par les espaces colorimétriques sRGB et Rec2020 intégrés avec les gamuts de certains autres espaces colorimétriques RVB bien connus.

Profils personnalisés

— **Profil de moniteur personnalisé** : Cette option vous permet de choisir un profil de moniteur personnalisé, par exemple si vous avez étalonné votre moniteur ou si vous utilisez un moniteur à large gammut. Un bouton permet d'effacer la sélection actuelle et une case à cocher permet de définir si le profil de moniteur personnalisé est actif ou non. S'il n'est pas actif, la transformation de l'affichage ciblera un affichage sRGB standard.

— **Intention de rendu** : Cette option vous permet de choisir votre intention de rendu ICC préférée, qui est utilisée pour la transformation des couleurs de l'espace de travail vers le profil de l'écran. Une description des intentions disponibles est disponible *sur la page Théorie*. Un bouton à cocher est également fourni pour définir si la compensation du point noir doit être utilisée ou non.

— **Profil d'épreuvage du dispositif de sortie** Cette option vous permet de choisir un profil d'épreuvage à l'écran pour un périphérique de sortie. Si aucun profil n'est défini ici, l'épreuvage à l'écran se fera avec le profil d'affichage. Un bouton permet d'effacer la sélection actuelle et une case à cocher permet d'activer ou non le profil d'épreuvage à l'écran. S'il n'est pas activé, le mode d'affichage de l'épreuvage à l'écran ne sera pas disponible.

Options d'importation/exportation de fichiers

— **Traiter les fichiers non gérés en couleur comme sRGB pour l'exportation vers les formats pris en charge** : Cette option concerne les images auxquelles aucun profil de couleur n'a été attribué. Si l'option est cochée, lorsqu'une image sans profil ICC est exportée vers un format qui prend en charge les profils ICC, elle sera enregistrée avec un profil ICC sRGB ou un profil Gris avec une CRT qui correspond au sRGB TRC standard. C'est essentiellement le comportement des versions précédentes de Siril. Il fonctionne souvent bien, mais ne représente pas un flux de travail correctement géré par la couleur. L'option est activée par défaut, mais il est préférable d'éviter ce problème en utilisant les options de conversion/assignation automatique pour s'assurer que les images auront un profil ICC assigné et qu'elles seront correctement gérées au moment où vous souhaitez les enregistrer.

— **Exporter les images 8 bits en utilisant** : Cette option définit l'espace colorimétrique à utiliser pour exporter les images 8 bits qui prennent en charge l'intégration des profils ICC : vous avez le choix entre sRGB et votre espace colorimétrique de travail tel qu'il a été choisi avec l'option Gamut RVB de travail ci-dessus.

— **Exporter des images 8 bits en utilisant :** Cette option définit l'espace colorimétrique pour l'exportation d'images 16 bits qui prennent en charge l'intégration de profils ICC : vous avez le choix entre sRGB et votre espace colorimétrique de travail tel que choisi avec l'option Espace colorimétrique préféré ci-dessus.

— **Intention d'exportation** : cela définit l'intention de transformation des images de leur espace colorimétrique attribué vers l'espace colorimétrique dans lequel elles doivent être enregistrées. Laissez ce paramètre sur Colorimétrie relative pour garantir que vos images enregistrées sont cohérentes avec la façon dont vous les avez enregistrées. D'autres intentions ne devraient probablement être choisies ici qu'à des fins de débogage et similaires.

Une préférence se trouve dans l'onglet Options FITS :

— **Intégrer les profils ICC dans les images FITS et les séquences FITSEQ** : Cette option permet l'intégration de profils ICC dans les fichiers FITS : les profils sont intégrés dans un HDU supplémentaire à l'intérieur du fichier FITS, avec le nom "ICCProfile", qui correspond à la méthode d'intégration utilisée par d'autres logiciels d'astrophotographie majeurs. *Il est fortement recommandé de laisser cette option activée.*

> **Astuce**
>
> L'exportation d'images vers des formats 8 bits et 16 bits qui ne prennent pas en charge les profils ICC intégrés se fera toujours dans l'espace colorimétrique sRGB.

Conversion automatique / Options d'affectation

Attribution automatique de l'espace colorimétrique de travail

Cet ensemble de cases à cocher définit les occasions au cours desquelles une image se verra automatiquement attribuer l'espace colorimétrique de travail. Les possibilités sont : au chargement, à l'empilement, à l'étirement et en sortie d'outil de composition (pixelmath / composition RVB).

La valeur par défaut consiste à attribuer l'espace colorimétrique de travail lors de l'étirement, car c'est à ce moment-là que vous passez généralement le mode de visualisation sur Linéaire et commencez à étirer votre image pour qu'elle ressemble à ce que vous souhaitez. C'est à ce moment-là que la gestion des couleurs commence à avoir de l'importance.

Cependant, vous pouvez également attribuer l'espace colorimétrique à des étapes antérieures du traitement si vous le souhaitez. Cela présente peu d'avantages ou d'inconvénients dans la plupart des cas, mais si vous prévoyez d'exporter des images avant de les étirer, cocher les autres cases garantira qu'un profil de couleur est toujours attribué.

> **Astuce**
>
> Un profil de couleur ne sera **jamais** automatiquement attribué si Siril pense qu'il s'agit d'un seul canal extrait d'une image tricolore en utilisant l'extraction RVB, TSV, TSL, CIE L*a*b* ou Vert. Il n'existe pas de profil ICC raisonnablement attribué dans ce cas. Vous devez considérer le fichier comme des données brutes jusqu'à ce que vous soyez prêt à le recomposer avec d'autres données ou à le traiter pour qu'il ait une belle apparence, auquel cas, vous devez attribuer votre espace colorimétrique de travail préféré.
>
> Les canaux CFA séparés, Ha et OIII peuvent se voir attribuer en toute sécurité une version linéaire de l'espace de travail, et cela sera fait si l'option rigoureuse décrite ci-dessous est active.

Si vous souhaitez éviter complètement les profils de couleur, vous pouvez sélectionner « Jamais », mais cela entraînera un flux de travail non géré par les couleurs et vos images risquent de ne pas paraître cohérentes lorsqu'elles sont visualisées dans d'autres logiciels ou lorsqu'elles sont sorties sur différents périphériques de sortie.

Attribuer de manière rigoureuse des profils ICC linéaires

Lors de l'attribution automatique de profils ICC à :
- des images chargées qui n'ont pas de profil ICC et ne montrent aucune preuve d'étirement précédent,
- images nouvellement empilées,
- images nouvellement composées

Siril attribuera un profil ICC linéaire. Ceci est généralement techniquement correct, car ces images contiennent toujours des données de lumière linéaire. L'option est disponible si vous le souhaitez, mais elle est souvent inutile : consultez la page *profiles* pour une explication de pourquoi. Vous souhaiterez peut-être l'utiliser afin de bénéficier de la transformation d'affichage en appliquant vos étirements aux données linéaires d'origine tout en prévisualisant l'effet en sRGB (ou votre profil de moniteur personnalisé) : dans ce cas, vous devez décocher la case de conversion automatique suivante, afin que le profil reste linéaire pendant l'étirement, puis convertissez manuellement les données en votre profil de travail une fois l'étirement terminé.

Convertir d'autres profils de couleur en profil de travail ?

Cette liste déroulante définit le comportement de Siril si une image est chargée avec un profil ICC différent du profil de travail sélectionné. Les options sont : Jamais, Toujours et Demander à chaque fois. La valeur par défaut est Jamais, car cela évite toute modification non demandée de l'image. Cependant, pour un flux de travail cohérent, vous souhaiterez peut-être sélectionner Toujours, afin de toujours modifier dans votre espace colorimétrique préféré. Demander à chaque fois est une option, mais cela peut être ennuyeux, surtout si l'on considère les profils d'exportation v2 comme étant différents des profils de travail intégrés v4, donc bien que cela soit prévu pour les utilisateurs qui souhaitent gérer chaque détail, ce n'est pas particulièrement recommandé.

Profils intégrés

Exporter les profils intégrés

Siril utilise des profils de la collection de profils de couleurs d'Elle Stone pour fournir ses profils intégrés. Comme ceux-ci sont distribués sous une licence Creative Commons BY-SA-3.0 qui nécessite le partage des profils, un bouton est prévu pour exporter tous les profils intégrés vers le répertoire de travail.

Afin de fournir un profil capable de l'intention Perceptuelle sRGB, le profil ICC Preference avec des tables de recherche perceptuelles est fourni. Pour l'exporter, vous devrez accepter les mêmes termes et conditions que pour le télécharger à partir du site Web de l'ICC.

4.1.9 Scripts

— L'onglet Scripts contient essentiellement les emplacements où Siril doit chercher les scripts. En effet, par défaut et en fonction du système d'exploitation utilisé, les scripts sont installés à un endroit spécifique :
  — `/usr/local/share/siril/scripts` ou `/usr/share/siril/scripts` sur GNU/Linux.
  — `C:\Program Files\Siril\scripts` sur Windows.
  — `/Applications/Siril.app/Contents/Resources/share/siril/scripts` sur MacOS, si l'application a été installée dans le dossier Applications.

> **Avertissement**
>
> Sur macOS, comme l'application est signée et notariée, il est impossible de modifier les scripts à l'intérieur du paquet. Sinon, l'application ne démarrera pas. Il faut donc définir un autre chemin pointant vers un dossier où vous avez les droits d'écriture.

— Le champ **Répertoire des scripts** permet de définir l'emplacement des dossier où ce trouvent les scripts que vous avez créé et/ou modifié. En cliquant sur le bouton juste en dessous, les dossiers seront à nouveau analysés et la liste des scripts dans le menu dédié sera mise à jour.
— La section **Avertissements** propose de désactiver :

1. Le texte d'avertissement qui s'affiche avant l'exécution d'un script.
2. La vérification du mot clé `requires` qui doit se trouver au tout début du script afin de vérifier si le script est compatible avec la version de Siril. Nous recommandons de ne pas décocher cette option.

— The **Siril Script Repository** field is where scripts that have been validated by the Siril team are automatically uploaded. However, these may have been written by third-party developers. You can enable/disable this feature with the *Enable use of the siril-scripts online repository* button, and define automatic or manual updates with *Automatic updates* and *Manual update*. To display a script in the script menu, simply select it.
— The **Reset python venv** will kill all running python scripts and reset the virtual environment "venv" directory to an as-new state. Any modules that have been installed by scripts will require re-installation. This is normally not necessary but is provided as a last resort to restore the venv to a pristine state as an aid to troubleshooting.

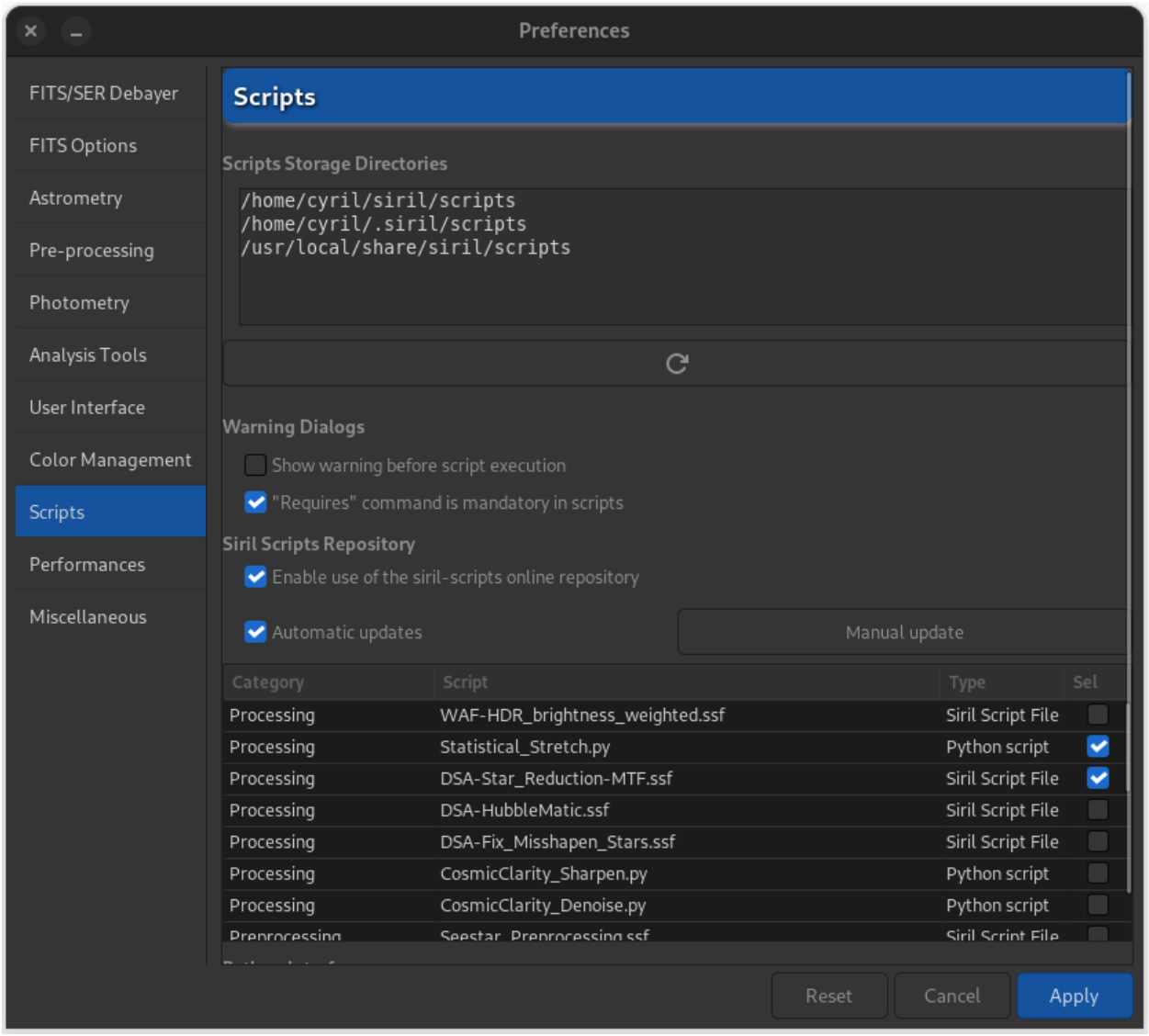


FIG. 12 – Page 8 de la boite de dialogue préférences

4.1.10 Performances

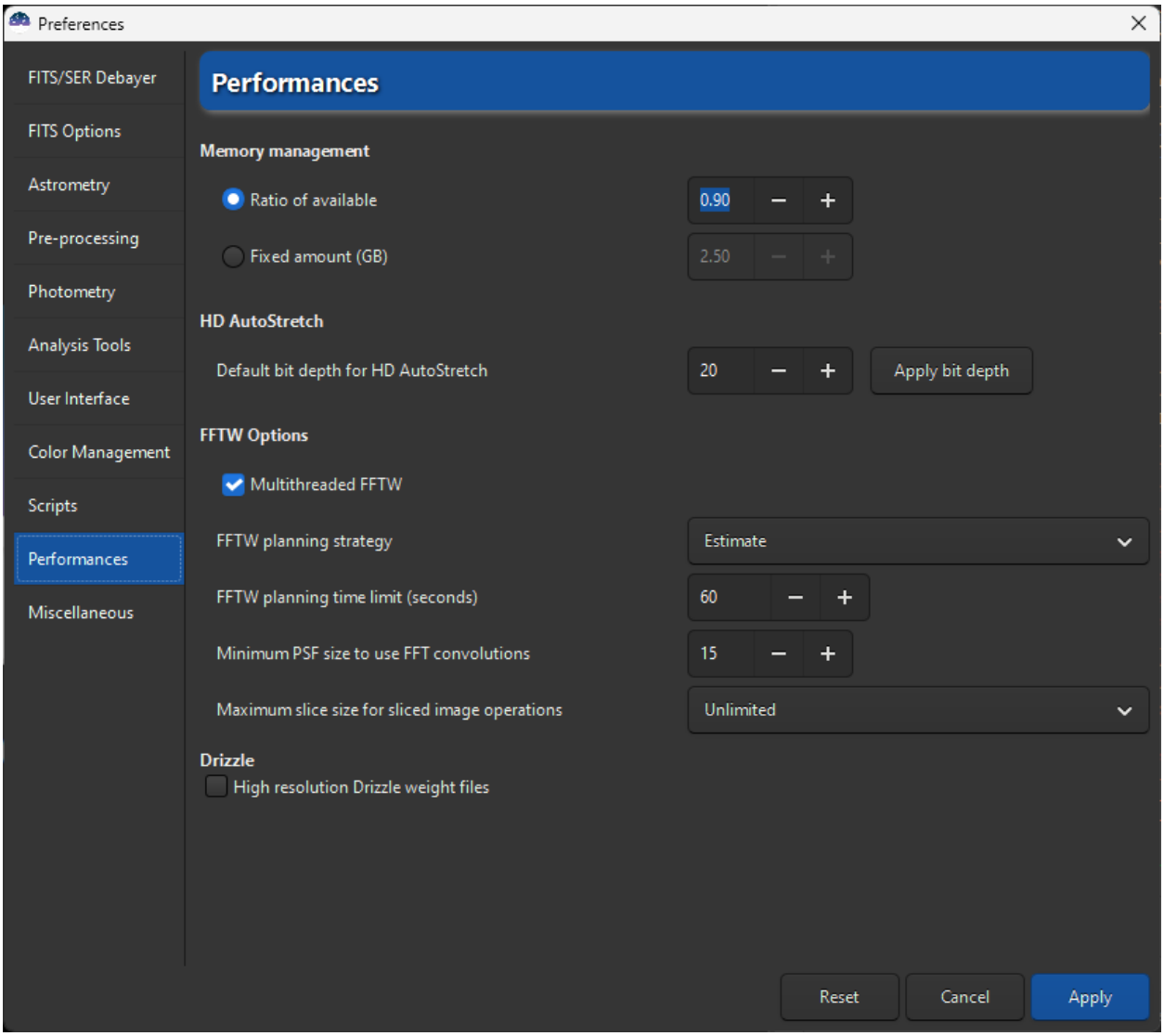


FIG. 13 – Page 9 de la boite de dialogue de préférences

Les logiciels de traitement d'images astronomiques, tels que Siril, utilisent beaucoup de ressources et nécessitent généralement des ordinateurs assez puissants. Il n'est pas impossible, lorsque l'ordinateur est très occupé, qu'il se bloque complètement. Il n'est pas du tout recommandé de faire quoi que ce soit d'autre sur l'ordinateur pendant le traitement, notamment de naviguer sur Internet, car les navigateurs sont très gourmands en mémoire vive. Il est cependant possible de gérer le pourcentage maximum de RAM que Siril peut utiliser.

— **Ratio de disponible** : Siril se limitera à un ratio de la quantité de mémoire physique libre et diminuera la taille des tâches si nécessaire. Une valeur supérieure à 1 signifie qu'une partie de la mémoire paginée sur une mémoire configurée sera utilisée, que le processus global sera plus lent et que le système ne répondra probablement pas pendant certaines opérations. Si vous n'avez pas configuré de mémoire paginée sur certains supports, une valeur de 1 ou plus entraînera probablement un blocage de siril ou du système d'exploitation.

— **Montant fixe (Go)** : Siril se limitera à une quantité fixe de mémoire et réduira la taille des tâches si nécessaire. La configuration d'une quantité de mémoire supérieure à celle disponible sur votre système peut entraîner un crache de Siril ou du système d'exploitation.

— L'option **Profondeur de bit par défaut pour l'auto-ajustement HD** définit la profondeur de bits par défaut pour

le mode d'affichage Auto ajustement HD. Des profondeurs de bits plus élevées nécessitent exponentiellement plus de mémoire pour la LUT et prennent plus de temps pour la recalculer, mais permettent de mieux lisser les artefacts de quantification lors de l'affichage d'images avec des pics d'histogramme très étroits. La profondeur de bits par défaut s'appliquera à partir du prochain changement de mode de visualisation, et peut être appliquée maintenant en utilisant le bouton à droite. Cliquez sur le bouton *Appliquer la profondeur de bit* pour définir la profondeur de bit de l'auto ajustement HD sélectionnée.

— **FFTW multithreadée** : ce bouton permet à FFTW (librairie de transformé de Fourier) d'utiliser plusieurs threads. Cela peut être plus rapide (bien que les performances n'augmentent pas linéairement avec le nombre de processeurs, en raison de la surcharge de synchronisation), mais la phase de planification de la FFTW prend plus de temps pour les systèmes multithreads, de sorte que la première FFTW pour une taille d'image donnée peut être considérablement plus lente en utilisant plusieurs threads.

— **Stratégie de planification FFTW** : cette liste déroulante définit la stratégie de planification de la FFTW. La librairie FFTW dispose de plusieurs algorithmes pour calculer une FFT et planifie une FFT donnée pour optimiser la vitesse. Elle enregistre les résultats de ces plans pour une réutilisation ultérieure dans un fichier cache appelé "Sagesse", de sorte qu'un peu de temps supplémentaire passé à planifier en amont peut être récompensé si vous calculez beaucoup de FFTs de la même taille. Notez que la sagesse est spécifique à une machine donnée : elle ne doit pas être partagée entre les machines et doit être supprimée et régénérée à partir de zéro après une mise à niveau de la mémoire ou du processeur ou un changement majeur de l'environnement logiciel (changements majeurs du système d'exploitation, changements majeurs de la version de Siril). Par ordre de rapidité, Estimer est la méthode la plus rapide : cette stratégie n'effectue aucune mesure mais planifie sur la base d'un ensemble d'heuristiques. Mesure est la deuxième méthode la plus rapide : cette méthode compare la vitesse des différentes méthodes internes de FFTW pour calculer la FFT et choisit la plus rapide. En conséquence, l'étape de planification prend plus de temps. Patient envisage encore plus de plans possibles, et Exhaustif en envisage encore plus. Si vous traitez toujours des images d'une taille spécifique, les stratégies de planification les plus coûteuses peuvent être intéressantes en raison de la sagesse, mais si vous travaillez avec des images de tailles différentes, un algorithme de planification moins coûteux peut s'avérer plus approprié.

— **Limite de temps de planification FFTW** : cette limite de temps arrête la planification FFTW après la limite de temps spécifiée. Ce délai est prioritaire sur la stratégie de planification. Notez que la limite de temps n'est pas strictement appliquée : La FFTW terminera tout calcul non interruptible qu'elle effectue au moment où la limite est atteinte, et si elle est fixée à zéro, la FFTW effectuera toujours au minimum la planification Estimate.

— **Drizzle : High resolution Drizzle weight files** : When this button is toggled, weight files produced by the drizzle algorithm will be of the same bitdepth than the images. This can be important when trying to get the best possible FWHM on the stacked image, but it is very much space consuming. By default, it is unticked and the weight files are saved in 8-bit.

4.1.11 Divers

Le dernier onglet contient tout ce qui n'entre pas dans les autres catégories.

— L'utilisation des boutons *Annuler*/*Rétablir* nécessite de l'espace disque. Beaucoup d'espace dans certains cas. Le dossier contenant les fichiers d'échange (qui sont les fichiers nécessaires au bon fonctionnement de la paire undo/redo) peut être défini dans la section **Répertoire swap**. L'espace disque est indiqué à droite du sélecteur de fichiers. Nous vous conseillons de **ne pas modifier** les paramètres par défaut, sauf si vous avez une bonne raison de le faire. Comme le choix d'un bon dossier est critique, il est possible de revenir au dossier par défaut en cliquant sur *Restaurer*.

— La partie **Avertissement** permet de désactiver certaines fenêtres d'avertissement qui sont là pour aider les débutants.

— **Astuces de l'introduction** : Au tout premier démarrage de Siril, il est possible de voir une petite animation montrant les nouveautés de l'application. Cette animation peut être rejouée en cliquant sur *Jouer l'introduction*.

— **StarNet Executable Location** : In order to use StarNet in Siril, it is necessary to tell 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.

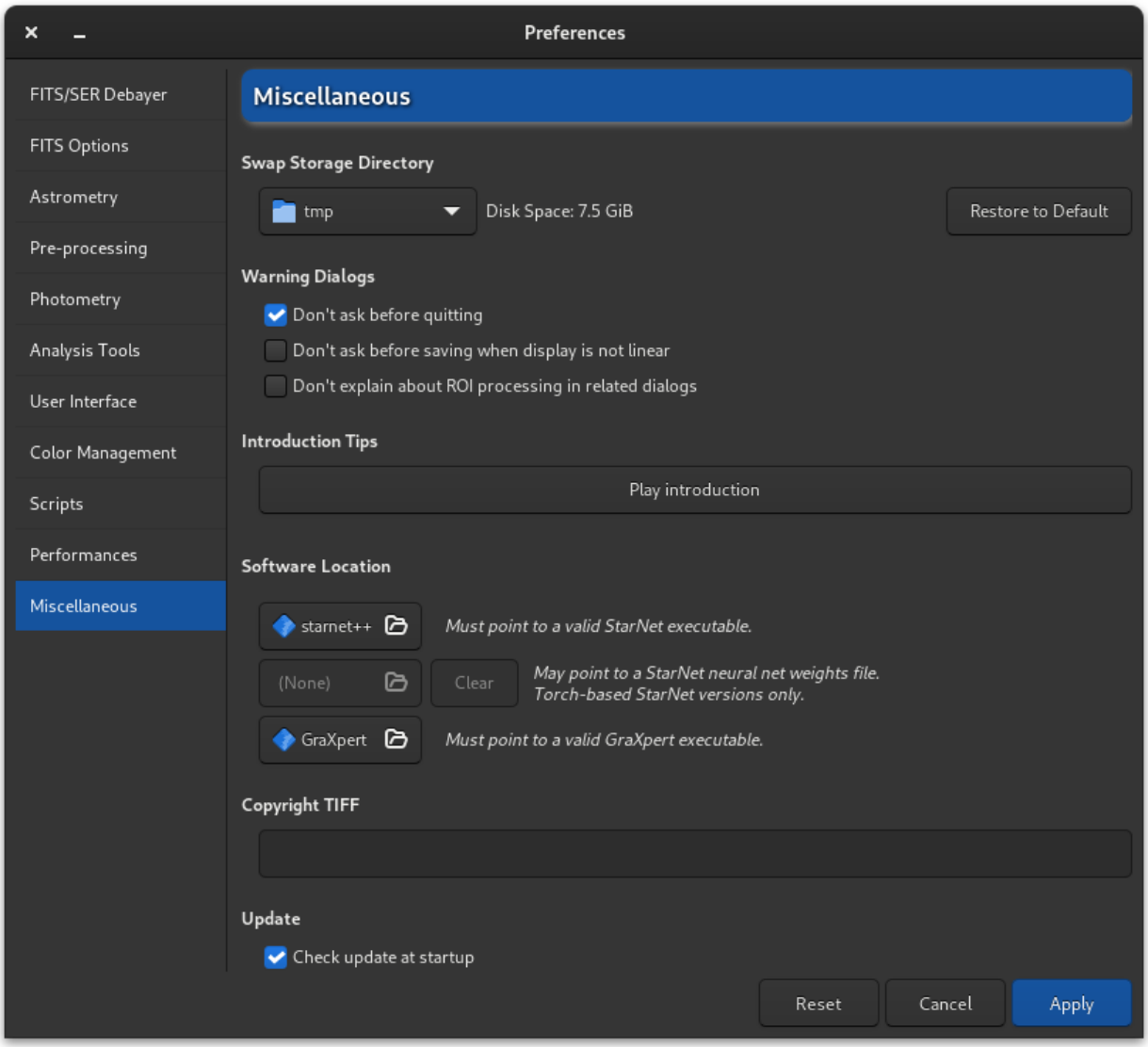


FIG. 14 – Page 10 de la boite de dialogue de préférences

— **Emplacement des poids de StarNet** : Les nouvelles versions de StarNet basées sur Torch offrent la possibilité d'indiquer l'emplacement d'un fichier de poids de réseau neuronal : il n'est pas nécessaire qu'il se trouve dans le même répertoire que l'exécutable. Cette préférence peut être utilisée pour définir l'emplacement d'un fichier de poids à transmettre à StarNet, et elle peut être réinitialisée à l'aide du bouton associé. Remarque : cette option ne fonctionne qu'avec les installations StarNet basées sur Torch. Avec les installations StarNet plus anciennes, le fichier de poids doit se trouver dans le même répertoire que l'exécutable.

Avertissement

Il s'agit de l'emplacement de la version de StarNet en ligne de commande qui doit être fournie, et non de la version graphique.

Avertissement

Sur macOS, il peut être difficile de trouver le chemin du répertoire car Apple ne facilite pas la navigation pour certains dossiers. Une astuce consiste à taper `Shift + Cmd + g` dans la boîte de dialogue de choix de fichier, puis d'entrer directement le chemin d'installation, qui est généralement celui défini par Homebrew. Habituellement, c'est `/usr/local/bin` sur les ordinateurs intel et `/opt/homebrew/bin/` sur les versions Apple Silicon.

— **GraXpert Executable Location** : In order to use GraXpert in Siril, it is necessary to tell Siril the path where the GraXpert executable is located. If you find any difficulty in getting GraXpert working with Siril, check that it works by itself before submitting a bug report : all Siril does is execute the third party software, so if it doesn't work by itself then it will not work when called from Siril either.
— **Copyright TIFF** : Lors de l'enregistrement de fichiers TIFF, il est possible de personnaliser le copyright dans les métadonnées EXIF dédiées.
— **Mise à jour** : Par défaut, Siril vérifie les mises à jour au démarrage. Vous êtes libre de désactiver ce comportement si vous ne souhaitez pas que l'application interroge notre site web.

4.2 Préférences (commandes)

À partir de la v1.2, la plupart des préférences peuvent également être définies par le biais de commandes, c'est-à-dire soit à partir d'une entrée directe dans la ligne de commande, soit par le biais de scripts ou en mode console.

Pour obtenir une liste de toutes les variables disponibles, utilisez la ligne de commande siril :

```
get -A
```

Ceci imprimera une liste de toutes les variables sur la console, avec leur valeur actuelle et une courte description (utilisez l'option minuscule *-a* pour omettre la description).

Le tableau ci-dessous en dresse la liste :

Variable	Défaut ([Plage])	Type	Commentaire
core.wd	(non défini)	répertoire	répertoire de travail actuel
core.extension	.fit	chaîne de caractères	extension du fichier FITS
core.force_16bit	faux	booléen	n'utilise pas 32 bits pour la profondeur de pixel

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
core.fits_save_icc	vrai	booléen	intégrer les profils ICC dans les fichiers FITS lors de l'enregistrement
core.allow_heterogeneous_fitseq	faux	booléen	permettre aux FITS cubes d'avoir des tailles différentes
core.mem_mode	0 [0, 1]	entier	mode mémoire (0 rapport, 1 montant)
core.mem_ratio	0.9 [0.05, 4]	double	ratio de la mémoire disponible
core.mem_amount	10 [0.1, 1e+06]	double	quantité de mémoire en GB
core.hd_bitdepth	20 [17, 24]	entier	Profondeur de bit de l'auto-ajustement HD
core.script_check_requires	vrai	booléen	la commande requires est nécessaire dans les scripts
core.pipe_check_requires	faux	booléen	besoin de la commande requires dans le tube nommé
core.check_updates	faux	booléen	vérifier les mise à jour au démarrage
core.lang	(non défini)	chaîne de caractères	langue de siril active
core.swap_dir	Dépendant de l'os	répertoire	répertoire d'échange
core.binning_update	vrai	booléen	mettre à jour la taille des pixels des images en binning
core.wcs_formalism	1 [0,1]	entier	formalisme WCS utilisé dans l'en-tête FITS
core.catalogue_namedstars	~/.local/share/kstars/	chaîne de caractères	Chemin d'accès du catalogue namedstars.dat
core.catalogue_unnamedstars	~/.local/share/kstars/	chaîne de caractères	Chemin d'accès du catalogue unnamedstars.dat
core.catalogue_tycho2	~/.local/share/kstars/	chaîne de caractères	Chemin d'accès du catalogue deepstars.dat
core.catalogue_nomad	~/.local/share/kstars/	chaîne de caractères	Chemin d'accès du catalogue USNO-NOMAD-1e8.dat
core.catalogue_gaia_astro	~/.local/share/siril/ga	chaîne de caractères	Chemin du catalogue astrométrique local Gaia
core.catalogue_gaia_photo	~/.local/share/siril/ga	chaîne de caractères	Chemin du catalogue photométrique local Gaia
core.rgb_aladin	faux	booléen	ajouter CTYPE3='RGB' dans l'en-tête FITS
core.use_checksum	faux	booléen	Vérifier les sommes de contrôle si elles existent
core.copyright	(non défini)	chaîne de caractères	copyright utilisateur à mettre dans l'en-tête du fichier
core.starnet_exe	(non défini)	chaîne de caractères	emplacement de l'exécutable StarNet

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
core.starnet_weights	(non défini)	chaîne de caractères	emplacement du fichier des poids de StarNet-torch
core.graxpert_path	(non défini)	chaîne de caractères	emplacement de l'exécutable GraXpert
core.asnet_dir	(non défini)	chaîne de caractères	répertoire de l'installation asnet_ansvr
core.fftw_timelimit	60	double	Délai de planification FFTW
core.fftw_conv_fft_cutoff	15	entier	Taille minimal du kernel de convolution pour utiliser FFTW
core.fftwf_strategy	0	entier	Stratégie de planification de la FFTW
core.fftw_multithreaded	vrai	booléen	FFTW multithreadée
core.max_slice_size	32769 [512, 32769]	entier	Maximum slice size for automated slice processing
starfinder.focal_length	0 [0, 999999]	double	longueur focale en mm pour le réglage du rayon
starfinder.pixel_size	0 [0, 99]	double	taille du pixel en µm pour le réglage du rayon
debayer.use_bayer_header	vrai	booléen	utiliser le motif de Bayer de l'en-tête du fichier
debayer.pattern	0 [0, 7]	entier	indice du motif Bayer
debayer.interpolation	8 [0, 9]	entier	type d'interpolation
debayer.orientation	0 [0, 3]	entier	préférence d'ordre des lignes
debayer.offset_x	0 [0, 1]	entier	Décalage en X de la matrice de Bayer
debayer.offset_y	0 [0, 1]	entier	Décalage en Y de la matrice de Bayer
debayer.xtrans_passes	1 [1, 4]	entier	Nombre de passes pour l'algorithme X-Trans Markesteijn
photometry.gain	2.3 [0, 10]	double	électrons par ADU pour l'estimation du bruit
photometry.inner	20 [2, 100]	double	rayon interne de l'anneau de fond de ciel
photometry.outer	30 [3, 200]	double	rayon externe de l'anneau de fond de ciel
photometry.inner_factor	4.2 [2, 50]	double	facteur de calcul automatique du rayon intérieur
photometry.outer_factor	6.3 [2, 50]	double	facteur de calcul automatique du rayon extérieur
photometry.force_radius	vrai	booléen	fixer la valeur d'ouverture du flux
photometry.auto_aperture_factor	4 [1, 5]	double	Rapport rayon/demi-FWHM
photometry.aperture	10 [1, 100]	double	ouverture fixe pour le calcul du flux
photometry.minval	-1500 [-65536, 65534]	double	valeur minimale valide du pixel pour la photométrie
photometry.maxval	60000 [1, 65535]	double	valeur maximale valide du pixel pour la photométrie
photometry.discard_var_catalogues	4 [0, 7]	entier	les catalogues à utiliser pour écarter les étoiles variables de la liste des étoiles de comparaison
photometry.redpref	(non défini)	chaîne de caractères	filtre rouge préféré pour SPCC
photometry.greenpref	(non défini)	chaîne de caractères	filtre vert préféré pour SPCC

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
photometry.bluepref	(non défini)	chaîne de caractères	filtre bleu préféré pour SPCC
photometry.lpfpref	(non défini)	chaîne de caractères	filtre LPF d'APN préféré pour SPCC
photometry.oscfilterpref	Aucun filtre	chaîne de caractères	filtre couleur préféré pour SPCC
photometry.monosensorpref	(non défini)	chaîne de caractères	capteur mono préféré pour SPCC
photometry.oscsensorpref	(non défini)	chaîne de caractères	capteur couleur préféré pour SPCC
photometry.is_mono	vrai	booléen	est-ce que le capteur pour SPCC est mono ?
photometry.is_dslr	faux	booléen	est-ce que le capteur couleur pour SPCC est d'APN ?
photometry.nb_mode	faux	booléen	sommes-nous en mode bande étroite ?
photometry.r_wl	656.28	double	longueur d'onde du filtre rouge
photometry.r_bw	6	double	largeur de bande du filtre rouge
photometry.g_wl	500.7	double	longueur d'onde du filtre vert
photometry.g_bw	6	double	largeur de bande du filtre vert
photometry.b_wl	500.7	double	longueur d'onde du filtre bleu
photometry.b_bw	6	double	largeur de bande du filtre bleu
astrometry.asnet_keep_xyls	faux	booléen	ne pas supprimer la table FITS .xyls
astrometry.asnet_keep_wcs	faux	booléen	ne pas supprimer le fichier de résultat .wcs
astrometry.asnet_show_output	faux	booléen	montrer la sortie de solve-field dans le log principal
astrometry.sip_order	3 [1, 5]	entier	degrés de la correction polynomiale
astrometry.radius	10 [0.01, 30]	double	rayon autour des coordonnées de la cible (degrés
astrometry.max_seconds_run	30 [0, 100000]	entier	maximum de secondes pour essayer de résoudre
astrometry.update_default_scale	vrai	booléen	mettre à jour la longueur focale et la taille de pixel par défaut à partir du résultat
astrometry.percent_scale_range	20 [1, 50]	entier	pourcentage en dessous et au-dessus de l'échantillonnage prévu
astrometry.default_obscode	(non défini)	chaîne de caractères	code de l'observatoire de l'UAI par défaut
analysis.panel	256 [127, 1024]	entier	taille du panneau de l'inspecteur d'aberration
analysis.window	381 [300, 1600]	entier	taille de la fenêtre de l'inspecteur d'aberration
compression.enabled	faux	booléen	Compression FITS activée
compression.method	0 [0, 3]	entier	méthode de compression FITS
compression.quantization	16 [8, 256]	double	facteur de quantification pour nombres flottants 32 bits
compression.hcompress_scale	4 [0, 256]	double	Facteur d'échelle HCompress
gui_prepro.cfa	faux	booléen	type de capteur pour la correction cosmétique

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
gui_prepro.equalize_cfa	vrai	booléen	égaliser les canaux du flat
gui_prepro.fix_xtrans	faux	booléen	activer la correction pour le capteur X-Trans
gui_prepro.xtrans_af_x	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_af_y	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_af_w	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_af_h	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_sample_x	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_sample_y	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_sample_w	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.xtrans_sample_h	0	entier	si aucun modèle X-Trans n'est trouvé, utilisez ceci
gui_prepro.bias_lib	(non défini)	chaîne de caractères	bias maître par défaut
gui_prepro.use_bias_lib	faux	booléen	utiliser le bias maître par défaut
gui_prepro.dark_lib	(non défini)	chaîne de caractères	master-dark par défaut
gui_prepro.use_dark_lib	faux	booléen	utiliser le dark maître par défaut
gui_prepro.flat_lib	(non défini)	chaîne de caractères	flat maître par défaut
gui_prepro.use_flat_lib	faux	booléen	utiliser le flat maître par défaut
gui_prepro.disto_lib	(non défini)	chaîne de caractères	master-distorsion par défaut
gui_prepro.use_disto_lib	faux	booléen	utiliser le fichier distorsion maître par défaut
gui_prepro.stack_default	$seq-name$stacked	chaîne de caractères	nom d'empilement par défaut
gui_prepro.use_stack_default	vrai	booléen	utiliser le nom de fichier empilé préféré
gui_registration.method	0 [0, 7]	entier	indice de la méthode d'alignement sélectionnée
gui_registration.interpolation	4 [0, 5]	entier	indice de la méthode d'interpolation sélectionnée
gui_registration.clamping	vrai	booléen	utiliser la méthode de contrainte avec les interpolations Lanczos et Cubique
gui_registration.drizz_weight_ma	faux	booléen	Match Drizzle weights bitpix to sequence
gui_stack.method	0 [0, 4]	entier	indice de la méthode sélectionnée
gui_stack.normalization	3 [0, 4]	entier	indice de la méthode de normalisation
gui_stack.rejection	5 [0, 7]	entier	indice de la méthode de rejet
gui_stack.weighting	0 [0, 4]	entier	indice de la méthode de pondération
gui_stack.sigma_low	3 [0, 20]	double	valeur sigma bas pour le rejet
gui_stack.sigma_high	3 [0, 20]	double	sigma valeur haute pour le rejet
gui_stack.linear_low	5 [0, 20]	double	valeur linéaire basse pour le rejet
gui_stack.linear_high	5 [0, 20]	double	valeur linéaire haute pour le rejet
gui_stack.percentile_low	3 [0, 100]	double	pourcentage valeur basse pour le rejet
gui_stack.percentile_high	3 [0, 100]	double	pourcentage valeur haute pour le rejet
gui.first_start	(non défini)	chaîne de caractères	premier démarrage de siril
gui.silent_quit	faux	booléen	ne pas confirmer la sortie en quittant

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
gui.silent_linear	faux	booléen	ne pas confirmer la sauvegarde en mode non linéaire
gui.remember_windows	vrai	booléen	se souvenir de la position de la fenêtre
gui.main_win_pos_x	0	entier	position principal de la fenêtre
gui.main_win_pos_y	0	entier	position principal de la fenêtre
gui.main_win_pos_w	0	entier	position principal de la fenêtre
gui.main_win_pos_h	0	entier	position principal de la fenêtre
gui.pan_position	-1	entier	position du séparateur des deux côtés
gui.extended	vrai	booléen	la fenêtre principale est étendue
gui.maximized	faux	booléen	la fenêtre principale est maximisée
gui.theme	0 [0, 1]	entier	indice du thème sélectionné
gui.font_scale	100	double	taille de la police en pourcentage
gui.icon_symbolic	faux	booléen	style d'icône
gui.script_path		liste des chaines	liste des répertoires de scripts
gui.use_scripts_repository	vrai	booléen	utiliser et synchroniser le dépôt de scripts en ligne
gui.use_spcc_repository	vrai	booléen	utiliser et synchroniser le dépôt de la base de données spcc
gui.auto_update_scripts	vrai	booléen	synchroniser automatiquement les dépôts de scripts en ligne
gui.auto_update_spcc	vrai	booléen	synchroniser automatiquement les dépôts de la base de données spcc
gui.selected_scripts		liste des chaines	liste des scripts sélectionnés dans le dépôt
gui.warn_scripts_run	vrai	booléen	avertissement lors de l'exécution d'un script
gui.show_thumbnails	vrai	booléen	montrer la vignette dans le sélecteur de fichiers
gui.thumbnail_size	256	entier	Taille des vignettes
gui.selection_guides	0	entier	nombre d'éléments des guides de la grille
gui.show_deciasec	faux	booléen	afficher les dixièmes d'arcsecondes au survol de l'écran
gui.default_rendering_mode	0 [0, 6]	entier	mode d'affichage par défaut
gui.display_histogram_mode	0 [0, 1]	entier	mode d'affichage de l'histogramme par défaut
gui.roi_mode	0	entier	Mode de sélection des ROI
gui.roi_warning	vrai	booléen	activer l'avertissement de la boîte de dialogue des ROI
gui.mmb_zoom_action	0	entier	Double clic sur le bouton du milieu de la souris pour zoomer
gui.mouse_speed_limit	0	double	Limite de vitesse du défilement fluide de la souris
gui.color_bkg_samples	rgba(255, 51, 26, 1.0)	chaîne de caractères	configurer la couleur des échantillons de l'extraction de gradient
gui.color_std_annotations	rgba(128, 255, 77, 0.9)	chaîne de caractères	configurer la couleur standard des annotations
gui.color_dso_annotations	rgba(255, 128, 0, 0.9)	chaîne de caractères	configurer la couleur des annotations du ciel profond

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
gui.color_sso_annotations	rgba(255, 255, 0, 0.9)	chaîne de caractères	configurer la couleur des annotations du système solaire
gui.color_tmp_annotations	rgba(255, 0, 0, 0.9)	chaîne de caractères	configurer la couleur des annotations temporaires
gui.custom_monitor_profile	(non défini)	chaîne de caractères	chemin d'accès au profil ICC du moniteur personnalisé
gui.soft_proofing_profile	(non défini)	chaîne de caractères	chemin vers l'épreuvage en douceur Profil de la CCI
gui.icc_custom_monitor_active	faux	booléen	profil de moniteur personnalisé actif
gui.icc_soft_proofing_active	faux	booléen	profil d'épreuvage de sortie actif
gui.custom_RGB_ICC_profile	(non défini)	chaîne de caractères	chemin d'accès au profil ICC RVB personnalisé
gui.custom_gray_ICC_profile	(non défini)	chaîne de caractères	chemin d'accès au profil ICC gris personnalisé
gui.rendering_intent	1	entier	intention de rendu de la gestion des couleurs
gui.proofing_intent	1	entier	intention de l'épreuvage de la gestion des couleurs
gui.export_intent	1	entier	color management export intent
gui.default_to_srgb	vrai	booléen	la valeur par défaut est sRGB lors de l'exportation d'images non gérées colorimétriquement
gui.working_gamut	0	entier	color management working gamut
gui.export_8bit_method	0	entier	color management export profile for 8bit files
gui.export_16bit_method	1	entier	color management export profile for 16bit files
gui.icc_autoconversion	0	entier	autoconvertir les images avec un profil ICC dans l'espace colorimétrique de travail
gui.icc_autoassignment	4	entier	encode les options d'attribution automatique des profils ICC
gui.icc_rendering_bpc	vrai	booléen	active le rendu BPC
gui.icc_pedantic_linear	faux	booléen	attribue de manière rigoureuse des profils ICC linéaires
gui.mouse_actions		liste des chaines	configuration des actions de la souris. Structure de données opaque, à éditer à l'aide de l'interface graphique
gui.scroll_actions		liste des chaines	configuration des actions de défilement. Structure de données opaque, à éditer à l'aide de l'interface graphique
gui_astrometry.compass_position	1 [0,5]	entier	indice de la position de la boussole sur la grille
gui_astrometry.cat_messier	vrai	booléen	montrer les objets Messier dans les annotations
gui_astrometry.cat_ngc	vrai	booléen	Affiche les objets NGC dans les annotations
gui_astrometry.cat_ic	vrai	booléen	montrer les objets IC dans les annotations
gui_astrometry.cat_ldn	vrai	booléen	Affiche les objets LDN dans les annotations
gui_astrometry.cat_sh2	vrai	booléen	Affiche les objets SH2 dans les annotations
gui_astrometry.cat_stars	vrai	booléen	afficher les étoiles dans les annotations
gui_astrometry.cat_const	vrai	booléen	afficher les constellations dans les annotations

suite sur la page suivante

Tableau 1 – suite de la page précédente

Variable	Défaut ([Plage])	Type	Commentaire
gui_astrometry.cat_const_names	vrai	booléen	afficher les noms des constellations dans les annotations
gui_astrometry.cat_user_dso	vrai	booléen	afficher les objets ciel profond de l'utilisateur dans les annotations
gui_astrometry.cat_user_sso	vrai	booléen	afficher les objets du système solaire de l'utilisateur dans les annotations
gui_pixelmath.pm_presets		liste des chaines	liste des préréglages du pixel math
script_editor.highlight_syntax	vrai	booléen	coloriser la syntaxe dans l'éditeur de script
script_editor.highlight_bracketma	vrai	booléen	mettre en évidence les parenthèses correspondantes dans l'éditeur de script
script_editor.rmargin	vrai	booléen	afficher la marge de droite dans l'éditeur de script
script_editor.rmargin_pos	80	entier	position de la marge droite dans l'éditeur de script
script_editor.show_linenums	vrai	booléen	afficher les numéros de ligne dans l'éditeur de script
script_editor.show_linemarks	faux	booléen	afficher les marqueurs de ligne dans l'éditeur de script
script_editor.highlight_currentlin	vrai	booléen	met en évidence la ligne en cours dans l'éditeur de script
script_editor.autoindent	vrai	booléen	indenter automatiquement les nouvelles lignes
script_editor.indentontab	vrai	booléen	indenter les blocs de lignes sélectionnés dans l'éditeur de script à l'aide de la touche de tabulation
script_editor.smartbs	vrai	booléen	Comportement intelligent du retour arrière dans l'éditeur de scripts
script_editor.smarthomeend	vrai	booléen	Smart Home / Comportement final dans l'éditeur de script
script_editor.showspaces	faux	booléen	Afficher les caractères d'espacement et de tabulation visibles dans l'éditeur de script
script_editor.shownewlines	faux	booléen	Afficher les caractères de retour à la ligne dans l'éditeur de script
script_editor.minimap	faux	booléen	Afficher une minimap dans l'éditeur de script

(*). Pour les catalogues kstars, il s'agit par défaut de `~/.local/share/kstars/`, quel que soit votre système d'exploitation.

Dans tous les cas, vous devrez les télécharger et définir le répertoire que vous avez choisi.

See section about using *local star catalogues*.

Les valeurs peuvent être récupérées avec la commande `get` :

Ligne de commande Siril

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

Obtient une valeur des paramètres en utilisant son nom, ou liste tout avec **-a** (liste de noms et de valeurs) ou avec **-A** (liste détaillée)

Voir aussi SET pour mettre à jour les valeurs

Liens : *set*

Les valeurs peuvent être modifiées avec la commande `set` :

Ligne de commande Siril

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

Met à jour une valeur de paramètre, en utilisant son nom de variable, avec la valeur donnée, ou un ensemble de valeurs en utilisant un fichier ini existant avec l'option **-import=**.
Voir GET pour obtenir des valeurs ou la liste des variables

Liens : *get*

CHAPITRE 5

Formats de fichier

Il existe plusieurs formats de fichiers que Siril peut ouvrir et avec lesquels il peut travailler. Cependant, seuls deux sont lus nativement et permettent de construire des *séquences* : les formats FITS et SER.

Nous examinerons ici les différents formats de fichiers lus par Siril et comprendrons les limites de certains et les points forts d'autres.

5.1 Profondeur de bits

La profondeur de bits définie la nombre de bits utiliser pour coder la couleur d'un seul pixel, ou le nombre de bits utilisé par chaque composante de couleur pour un seul pixel.

Pour les images de la vie quotidienne, 8 bits suffisent amplement. Cela signifie qu'un pixel est codé sur des valeurs comprises dans l'intervalle [0, 255]. Toutefois, la photographie d'objets astronomiques est plus exigeante et nécessite généralement de travailler sur des images d'une profondeur de bits d'au moins 16 bits : c'est-à-dire dans la plage [0, 65535]. Mieux encore, une précision de 32 bits permet de conserver les informations les plus subtiles. Sur ce dernier type, les pixels sont soit encodés dans l'intervalle [0, 4294967295], soit, comme utilisé dans Siril, entre les valeurs flottantes [0, 1]. Il est possible de trouver des formats codant les pixels sur 64 bits (dans l'intervalle [0, 1], mais ils sont rares et ont une utilisation très spécifique. Le format FITS notamment le permet.

Cependant, tous les formats de fichiers d'images ne supportent pas le 16 bits, et encore moins le 32 bits. Il faut donc en tenir compte lors du choix du format.

5.2 Format de fichier commun

Les formats de fichiers d'images présentés ici sont des formats standard, lisibles par tous les logiciels de traitement d'images. Ces formats ont été conçus pour répondre à des besoins anciens et peuvent être obsolètes. De plus, aucun de ces formats n'a été conçu pour traiter des données astronomiques. Ils doivent donc généralement être utilisés en fin de chaîne de traitement.

FIG. 1 – Image linéaire sauvegardée en 16 bits

FIG. 2 – La même image linéaire sauvegardée en 8 bits. Presque toutes les données ont été perdues

5.2.1 Format AVIF

AVIF est un format d'image moderne basé sur le codec vidéo AV1. Ce format utilise le format de conteneur HEIC. Siril supporte la lecture d'images AVIF jusqu'à une profondeur de 12 bits. Un support sans perte peut être disponible en fonction des codecs compilés dans la librairie libheif installée. AVIF supporte les profils de couleur utilisant à la fois les profils ICC intégrés et les identifiants NCLX : les profils ICC intégrés sont entièrement supportés, et il y a un support partiel pour les identifiants NCLX (ils sont convertis en profil ICC équivalent, bien qu'actuellement seuls les profils NCLX les plus probables soient supportés).

Les extensions de nom de fichier pour ce format sont **.avif**.

5.2.2 Format BMP

Les fichiers portant l'extension **.bmp** sont des fichiers d'images bitmap utilisés pour stocker des images bitmap numériques. Ces images sont indépendantes de la carte graphique et sont également appelées format de fichier DIB (Device Independent Bitmap). Cette indépendance permet au fichier d'être ouvert sur de multiples plateformes telles que Microsoft Windows et Mac. Le format de fichier BMP permet de stocker des données sous forme d'images numériques bidimensionnelles, à la fois en monochrome et en couleur, avec différentes profondeurs de couleur.

Aujourd'hui, ce format n'est plus vraiment utilisé et d'autres types de fichiers sont préférés.

5.2.3 Format HEIF

HEIF est un format d'image moderne basé sur le codec vidéo x265. Ce format utilise le format de conteneur HEIC. Siril supporte la lecture d'images HEIC jusqu'à une profondeur de 12 bits. Un support sans perte peut être disponible en fonction des codecs compilés dans la libheif installée. HEIC supporte les profils de couleur utilisant à la fois les profils ICC intégrés et les identifiants NCLX : les profils ICC intégrés sont entièrement supportés, et il y a un support partiel pour les identifiants NCLX (ils sont convertis en profil ICC équivalent, bien qu'actuellement seuls les profils NCLX les plus probables soient supportés).

Les extensions de nom de fichier pour ce format sont **.heic** et **.heif**.

5.2.4 Format JPEG

Probablement le format de fichier le plus utilisé pour le partage d'images sur les forums, par e-mail ou sur des clés usb. Ce format permet une compression plus ou moins forte (destructive) qui donne des tailles de fichiers idéales pour les échanges. L'extension de ce type de fichier est **.jpg** ou **.jpeg**.

Le format JPEG n'est cependant codé qu'en 8 bits. Avec la compression qui produit des artefacts, ce format n'est pas très adapté aux images d'astronomie et on lui préfère généralement le format PNG.

5.2.5 Format JPEG XL

Après plusieurs mises à jour du format JPEG (JPEG 2000, JPEG XR...), JPEG XL représente un saut majeur dans les capacités avec l'intention déclarée de remplacer tous les formats matriciels. Qu'il y parvienne ou non, il présente plusieurs caractéristiques qui le rendent intéressant pour l'astrophotographie et Siril a donc ajouté la prise en charge de la lecture et de l'écriture. Il prend en charge des profondeurs de bits élevées, jusqu'à 32 bits flottants. Il prend en charge la compression avec et sans perte : pour la compression avec perte, il permet d'obtenir des taux de compression bien supérieurs à ceux du JPEG pour le même paramètre de qualité, et pour la compression sans perte, il permet d'obtenir des taux de compression bien supérieurs à ceux du PNG. Notez que les métadonnées FITS ne seront pas préservées lors de l'enregistrement en JPEG XL : son utilisation principale dans Siril sera probablement un format d'archivage pour les rendus finaux d'images. Il peut également être utile pour l'exportation vers le web : le format n'est pas encore largement supporté par les navigateurs, mais Safari le supporte nativement et le support est disponible expérimentalement ou à l'aide de modules complémentaires dans Firefox et Chrome, de sorte que nous pourrions voir le support se généraliser à l'avenir.

JPEG XL prend entièrement en charge les profils ICC intégrés, mais pour améliorer la compression, il enregistre les profils qu'il reconnaît avec un identifiant codé au lieu du profil complet. Cela signifie que la description du profil ICC

que vous voyez après avoir ouvert un fichier JPEG XL sauvegardé peut ne pas être la même qu'à l'origine, mais le profil ICC sera fonctionnellement identique.

L'extension du nom de fichier pour ce format est **.jxl**.

5.2.6 Format PNG

Portable Network Graphics est un format de fichier graphique matriciel qui prend en charge la compression des données sans perte. L'extension du format est **.png**. Les images PNG en niveaux de gris prennent en charge la plus large gamme de profondeurs de pixels de tous les types d'images. Les profondeurs de 1, 2, 4, 8 et 16 bits sont prises en charge, couvrant tout des simples scans en noir et blanc aux images médicales en profondeur et aux images astronomiques brutes.

Les données d'images astronomiques calibrées sont généralement stockées sous forme de valeurs à virgule flottante de 32 ou 64 bits, et certaines données brutes sont représentées sous forme d'entiers de 32 bits. Aucun des deux formats n'est directement pris en charge par PNG.

Néanmoins, ce format est un excellent choix pour sauvegarder l'image finale, après traitement.

5.2.7 Format TIFF

Le format TIFF ou TIF (Tagged Image File Format) représente des images matricielles destinées à être utilisées sur divers appareils conformes à cette norme de format de fichier. Il est capable de décrire des données d'image à deux niveaux, en niveaux de gris, en couleurs de palette et en couleurs complètes dans plusieurs espaces colorimétriques. Il prend en charge les schémas de compression avec et sans perte pour permettre aux applications qui utilisent ce format de choisir entre l'espace et le temps. L'extension est soit **.tiff**, soit **.tif**.

Les avantages du format TIFF sont multiples. Il supporte un encodage jusqu'à 32 bits par pixels et offre une grande variété de champs possibles dans les métadonnées, ce qui en fait un bon candidat pour le stockage des données astronomiques.

En utilisant le format TIFF, et en collaboration avec d'autres développeurs, nous avons mis en place une pseudo norme, *Astro-TIFF*.

5.2.8 Format NetPBM

Plusieurs formats graphiques sont utilisés et définis par le projet Netpbm. Les formats portable pixmap (PPM), portable graymap (PGM) et portable bitmap (PBM) sont des formats de fichiers d'images conçus pour être facilement échangés entre plates-formes. Les extensions de fichiers possibles sont **.pbm**, **.pgm** (pour les fichiers en niveaux de gris) et **.ppm**.

Ces formats, qui prennent en charge jusqu'à 16 bits par canal, sont peu utilisés et ne devraient être utilisés que pour le stockage final des images.

5.2.9 Format AVI

Il s'agit d'un format de film, capable de contenir des données avec différents codecs audio et vidéo. Il existe des codecs vidéo sans perte qui ont été utilisés dans le passé pour l'imagerie astronomique, mais il s'agit d'un format qui ne contient pas de métadonnées utilisables pour l'astronomie, qui est limité aux images 8 bits et qui ne garantit pas que les données qu'il contient soient brutes.

Avertissement

Ce format de fichier d'entrée est maintenant déprécié. Nous vous recommandons d'utiliser le format *SER* à la place.

5.3 FITS

5.3.1 Spécification

FITS signifie **Flexible Image Transport System** et est un standard de format de données astronomiques utilisé par les scientifiques professionnels tel que la NASA. FITS est bien plus qu'un format d'image (comme JPG ou TIFF) et est principalement conçu pour stocker des données scientifique dans des tableaux multidimensionnels.

Un fichier FITS se compose d'une ou plusieurs unités d'en-tête et de données (HDU), le premier HDU étant appelé "HDU primaire" ou "tableau primaire". Cinq types de données primaires sont pris en charge : octets non signés de 8 bits, entiers signés de 16 et 32 bits et réels flottants de 32 et 64 bits en simple et double précision. Le format FITS peut également stocker des entiers non signés de 16 et 32 bits.

Chaque unité d'en-tête est constituée d'un nombre quelconque de mots-clés enregistrés de 80 caractères qui ont la forme générale suivante :

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

Les noms de mots-clés peuvent comporter jusqu'à 8 caractères et ne peuvent contenir que des lettres majuscules, des chiffres de 0 à 9, trait d'union et caractère de soulignement. Le nom du mot-clé est (généralement) suivi d'un signe égal et d'un espace (=) dans les colonnes 9 et 10 de l'en-tête, puis de la valeur du mot-clé, qui peut être un nombre entier, un nombre à virgule flottante, une chaîne de caractères (entre guillemets simples) ou une valeur booléenne (la lettre **T** ou **F**).

Le dernier mot-clé de l'en-tête est toujours le mot-clé `END` qui n'a pas de champ de valeur ou de commentaire.

Chaque unité d'en-tête commence par une série de mots-clés obligatoires qui spécifient la taille et le format de l'unité de données qui suit. L'en-tête principale d'une images à deux dimensions, par exemple, commence par les mots-clés suivants :

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

Note

Dans Siril, les fichiers FITS 64 bits ne sont pas supportés. Siril les lit mais les convertit en fichiers 32 bits.

5.3.2 Nom de fichier

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

Avertissement

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

5.3.3 Compression

La compression est le moyen de réduire la taille de l'image. Il existe de nombreuses méthodes de compression en fonction du type d'images utilisées. Cette compression peut être destructive, comme avec le JPEG, ou sans perte comme le propose le PNG.

Il est possible de travailler avec des fichiers FITS compressés. Au prix d'un temps de calcul plus long, la taille des images peut être considérablement réduite. Siril propose plusieurs algorithmes de compression qui sont les suivants :

- **Rice** : L'algorithme Rice est simple et très rapide
- **GZIP 1** : L'algorithme gzip est utilisé pour compresser et décompresser les pixels de l'image. Gzip est l'algorithme de compression utilisé dans le logiciel libre GNU du même nom.
- **GZIP 2** : Les octets du tableau des valeurs des pixels de l'image sont mélangés par ordre décroissant de signification avant d'être compressés à l'aide de l'algorithme gzip. Cette méthode est particulièrement efficace lors de la compression de tableaux à virgule flottante.

Une option est associée à ces algorithmes, le **Niveau de quantification** :

Bien que les images en virgule flottante puissent être compressées sans perte (à l'aide de gzip, puisque Rice ne compresse que les tableaux d'entiers), ces images ne se compressent souvent pas très bien parce que les valeurs des pixels sont trop bruitées ; les bits de poids faible de la mantisse des valeurs des pixels contiennent en effet des motifs de bits aléatoires incompressibles. Pour obtenir une meilleure compression, il faut supprimer une partie de ce bruit, mais sans perdre l'information utile qu'il contient. Si elle est trop grande, on sous-échantillonne les valeurs des pixels, ce qui entraîne une perte d'informations dans l'image. En revanche, s'il est trop petit, il préserve une trop grande partie du bruit (voire l'amplifie) dans les valeurs des pixels, ce qui se traduit par une mauvaise compression.

> **Note**
>
> Les algorithmes de compression d'image supportés sont tous **sans perte** lorsqu'ils sont appliqués à des images FITS entières ; les valeurs des pixels sont préservées exactement sans perte d'information pendant le processus de compression et de décompression. Les images FITS à virgule flottante (qui ont `BITPIX = -32` ou `-64`) sont d'abord quantifiées en valeurs de pixels entiers mis à l'échelle avant d'être compressées. Cette technique permet d'obtenir des facteurs de compression beaucoup plus élevés qu'en utilisant simplement GZIP pour compresser l'image, mais cela signifie également que les valeurs flottantes originales des pixels peuvent ne pas être restituées avec précision lorsque l'image est décompressée. Lorsque cette technique est utilisée correctement, elle ne fait qu'éliminer le "bruit" des valeurs à virgule flottante sans perdre d'informations significatives.

5.3.4 Cas particuliers des fichiers scientifiques FITS

Les fichiers FITS produits par les grandes missions astronomiques telles que celles du télescope spatial Hubble (HST) de la NASA ou du télescope spatial James Webb (JWST) sont souvent plus complexes que les fichiers FITS typiques utilisés par les astronomes amateurs. Ces fichiers sont généralement structurés comme des séquences FITS (également appelées cubes FITS), mais avec une différence importante : les images individuelles à l'intérieur de ces séquences peuvent avoir des dimensions différentes.

Cette fonction permet de stocker des données multidimensionnelles collectées dans des conditions variables, où chaque image de la séquence peut ne pas avoir la même résolution ou la même taille. Par exemple, les données du JWST comportent souvent des ensembles de données hétérogènes en raison de la nature de ses instruments et de ses modes d'observation.

Dans Siril, lors du traitement de ces types de fichiers FITS scientifiques, il est crucial d'activer l'option qui permet de traiter des images de dimensions différentes dans un cube FITS. Ceci peut être fait en sélectionnant l'option *Autoriser les cubes FITS à avoir des images de tailles différentes* dans l'onglet *Options FITS* des préférences.

Avertissement

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

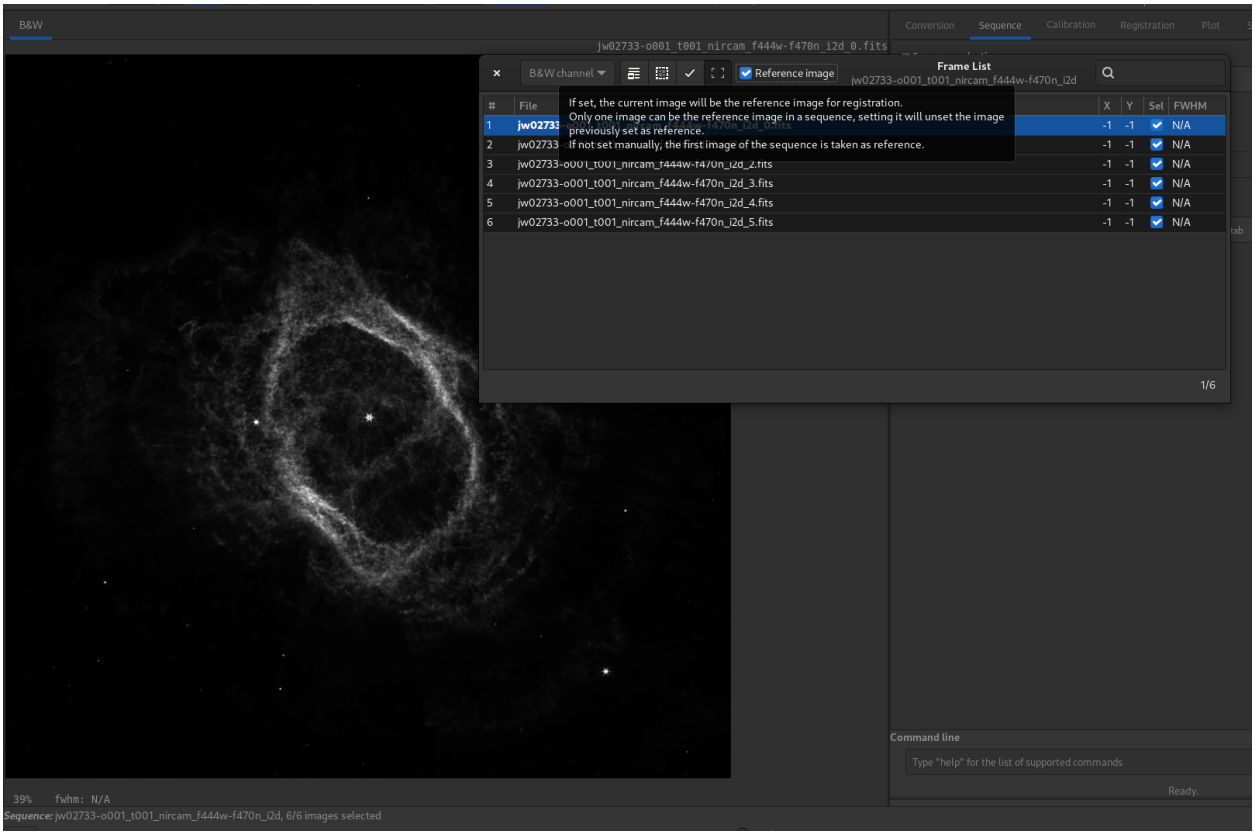


FIG. 3 – Exemple d'une image du JWST chargée dans Siril

5.3.5 Orientation des images FITS

Le standard FITS est un conteneur qui décrit comment stocker des données d'image et des métadonnées. Les outils professionnels, depuis les débuts du format FITS, comme ds9 (Harvard Smithsonian Center for Astrophysics), fv (FITS viewer de la NASA), stockent les images **de bas en haut**. Nous pourrions être tentés de dire que cela n'a pas vraiment d'importance, mais lorsque le dématriçage ou l'astrométrie sont impliqués, des problèmes se posent. Par exemple, le motif de Bayer **RVVB** habituel devient **VBRV** si l'image est à l'envers.

Aujourd'hui, malgré cela, la plupart des pilotes des caméras écrivent les données dans l'ordre descendant et nous devons nous en accommoder.

Pour ces raisons, nous avons récemment introduit, avec P. Chevalley de CCDCiel, un **nouveau mot-clé FITS**. Nous encourageons tous les producteurs de données, les développeurs INDI et ASCOM, à l'utiliser afin de faciliter les choses pour tout le monde.

Ce mot clé est `ROWORDER` de type `TSTRING`. Il peut prendre deux valeurs : `BOTTOM-UP` et `TOP-DOWN`.

Siril lit et affiche toujours les images dans l'ordre ascendant, mais si l'information descendante est spécifiée dans le mot-clé, alors Siril dématrice l'image avec le motif corrigé.

Pourquoi certains programmes écrivent-ils les images de manière ascendante ?

La raison en est la suivante : les mathématiques le font de cette façon.

De plus, la spécification FITS précise :

> ***5.1. Conventions d'affichage des images***
>
> *Il est très utile d'adopter une convention pour l'affichage des images transférées via le format FITS. De nombreux systèmes actuels de traitement d'images ont convergé vers une telle convention. Nous recommandons donc aux auteurs de FITS d'ordonner les pixels de manière à ce que le premier pixel du fichier FITS (pour chaque plan d'image) soit celui qui serait affiché dans le coin inférieur gauche (avec le premier axe croissant vers la droite et le deuxième axe croissant vers le haut) par le système d'imagerie de l'auteur de FITS. Cette convention est clairement utile en l'absence d'une description des coordonnées mondiales. Elle n'empêche pas un programme de consulter les descriptions des axes et d'outrepasser cette convention, ni l'utilisateur de demander un affichage différent. Cette convention ne dispense pas non plus les auteurs de FITS de fournir des descriptions complètes et correctes des coordonnées de l'image, permettant ainsi à l'utilisateur de déterminer la signification de l'image. L'ordre d'affichage de l'image est simplement une convention de commodité, alors que les coordonnées des pixels font partie de la physique de l'observation.*

Avertissement

Le mot-clé `ROWORDER` peut être utilisé pour :

1. Affichage de l'image avec l'orientation prévue (afficher une image non inversée).
2. Retourner le motif de matrice de Bayer. Le motif de Bayer peut donc être celui spécifié conforme par le fournisseur du capteur.

MAIS

1. `ROWORDER` ne doit pas être utilisé pour déplier les données de l'image pour l'empilement. Sinon, les nouvelles images deviendraient incompatibles avec les anciennes images darks et flats.
2. `ROWORDER` ne doit pas être utilisé pour déplier les données de l'image pour la résolution astrométrique. Cela rendrait la solution astrométrique incompatible avec d'autres programmes.

5.3.6 Logiciel utilisant le mot-clé `ROWORDER`

— Siril (depuis la version 0.99.4)
— CCDCiel (depuis la version 0.9.72)
— Indi (depuis Jul. 2020)
— KStars (depuis 3.4.3)
— SharpCap (depuis la version 3.3)
— FireCapture (depuis la version 2.7)
— N.I.N.A (depuis la version 1.10)
— MaxImDL (depuis la version 6.23)
— INDIGO (depuis Jul. 2020)
— PixInsight (depuis la version 1.8.8-6)
— ASTAP (depuis la version ß0.9.391)
— APT (depuis la version 3.86.3)
— AstroDMx Capture (depuis la version 0.80)
— Astroart (depuis la version 8.0)

5.3.7 Récupérer la matrice de Bayer

L'ordre des lignes de l'image change la façon dont la matrice de Bayer doit être lue, mais il y a aussi deux mots-clés optionnels dans l'en-tête FITS qui ont un effet sur cela : `XBAYROFF` et `YBAYROFF`. Ils spécifient un décalage par rapport à la matrice de Bayer, pour commencer à la lire sur la première colonne ou la première ligne.

Pour aider les développeurs à intégrer les mots-clés `ROWORDER`, `XBAYROFF` et `YBAYROFF` dans leurs logiciels, Han Kleijn de hnsky.org a créé des images de test, une pour chaque combinaison des trois mots-clés. Téléchargez-les ici : Bayer_test_pattern_v6.tar.gz.

5.3.8 Liste des mots-clés FITS

Siril peut lire et interpréter un large éventail de mots-clés. La liste suivante illustre les mots-clés non standard que le Siril enregistre si nécessaire. Certains mots-clés lus par le Siril peuvent ne pas apparaître dans cette liste. Par exemple, les mots-clés `CCDTEMP` ou `TEMPERAT`, qui indiquent la température du capteur, sont correctement lus, mais sont propagés sous le mot-clé `CCD-TEMP`.

Astuce

Siril is able to read and compare checksums if they are present in the FITS header. However, by default and for software performance reasons, `CHECKSUM` and `DATASUM` cards are automatically removed from HDU headers when a file is opened, and any `CHECKSUM` or `DATASUM` cards are stripped from headers when an HDU is written to a file. Nevertheless, they can be saved at the user's request, using the **-chksum** option of the *save* command, or via the graphical user interface.

Tableau 1: Mots-clés FITS enregistrés par Siril. Pour des raisons de clarté, les mots-clés SIP de l'astrométrie ne sont pas listés.

Mots clés FITS	Type	Commentaire
BZERO	Double	Décalage de la plage de données par rapport à celle d'une valeur non signée (unsigned short)
BSCALE	Double	Facteur d'échelle par défaut
MIPS-HI	Unsigned short	Coupure supérieure de visualisation
MIPS-LO	Unsigned short	Coupure inférieure de visualisation
MIPS-FHI	Float	Coupure supérieure de visualisation
MIPS-FLO	Float	Coupure inférieure de visualisation
PROGRAM	String	Logiciel qui a créé ce HDU
FILENAME	String	Nom de fichier d'origine
DATE	String	Date UTC de création du fichier FITS
DATE-OBS	String	YYYY-MM-DDTh :mm :ss début de l'observation, UT
IMAGETYP	String	Types d'image
ROWORDER	String	Ordre des lignes dans le tableau d'images
EXPTIME	Double	[s] Durée du temps d'exposition
TELESCOP	String	Télescope utilisé pour acquérir cette image
OBSERVER	String	Nom de l'observateur
FILTER	String	Nom du filtre actif
APERTURE	Double	Ouverture de l'instrument
ISOSPEED	Double	Réglage ISO de l'appareil photo
FOCALLEN	Double	[mm] Longueur focale
CENTALT	Double	[deg] Altitude du télescope
CENTAZ	Double	[deg] Azimut du télescope
XBINNING	Unsigned int	Mode de binning de la caméra

suite sur la page suivante

Tableau 1 – suite de la page précédente

Mots clés FITS	Type	Commentaire
YBINNING	Unsigned int	Mode de binning de la caméra
XPIXSZ	Double	[um] Taille du pixel en X
YPIXSZ	Double	[um] Taille du pixel en Y
INSTRUME	String	Nom de l'instrument
CCD-TEMP	Double	[degC] température CCD
SET-TEMP	Double	[decC] Température de consigne du CCD
GAIN	Unsigned short	Gain du capteur
OFFSET	Unsigned short	Offset du gain du capteur
CVF	Double	[e-/ADU] Électrons par unité A/D
BAYERPAT	String	Matrice de Bayer
XBAYROFF	Int	Décalage X de la matrice de Bayer
YBAYROFF	Int	Décalage Y de la matrice de Bayer
FOCNAME	String	Nom de l'équipement de mise au point
FOCPOS	Int	[pas] Position du focuseur
FOCUSSZ	Int	[um] Taille du pas du focuseur
FOCTEMP	Double	[degC] Température du focuseur
STACKCNT	Unsigned int	Images d'empilement
LIVETIME	Double	[s] Temps d'exposition après correction du temps mort
EXPSTART	Double	[JD] Heure de début de l'exposition (date julienne standard)
EXPEND	Double	[JD] Heure de fin d'exposition (date julienne standard)
OBJECT	String	Nom de l'objet d'intérêt
AIRMASS	Double	Masse d'air au centre du cadre (Gueymard 1993)
SITELAT	Double	[deg] Latitude du site d'observation
SITELONG	Double	[deg] Longitude du site d'observation
SITEELEV	Double	[m] Élévation du site d'observation
DFTTYPE	String	Module/phase d'une transformée de Fourier discrète
DFTORD	String	Les fréquences spatiales basses/hautes sont situées au centre de l'image
DFTNORM1	Double	Valeur de normalisation pour le canal #1
DFTNORM2	Double	Valeur de normalisation pour le canal #2
DFTNORM3	Double	Valeur de normalisation pour le canal #3
CTYPE3	String	Image RGB
OBJCTRA	String	Ascension droite du centre de l'image (hms)
OBJCTDEC	String	Déclinaison du centre de l'image (dms)
RA	Double	Ascension droite du centre de l'image (deg)
DEC	Double	Déclinaison du centre de l'image (deg)
CTYPE1	String	Projection TAN (gnomique)
CTYPE2	String	Projection TAN (gnomique)
CUNIT1	String	Projection TAN (gnomique) + distorsions SIP
CUNIT2	String	Projection TAN (gnomique) + distorsions SIP
EQUINOX	Double	Equinoxe équatorial
CRPIX1	Double	Pixel de référence de l'axe 1
CRPIX2	Double	Pixel de référence de l'axe 2
CRVAL1	Double	Valeur de référence de l'axe 1 (deg)
CRVAL2	Double	Valeur de référence de l'axe 2 (deg)
LONPOLE	Double	Longitude native du pôle céleste
CDELT1	Double	Taille du pixel en X (deg)
CDELT2	Double	Taille du pixel en Y (deg)
PC1_1	Double	Matrice de transformation linéaire (1, 1)

suite sur la page suivante

Tableau 1 – suite de la page précédente

Mots clés FITS	Type	Commentaire
PC1_2	Double	Matrice de transformation linéaire (1, 2)
PC2_1	Double	Matrice de transformation linéaire (2, 1)
PC2_2	Double	Matrice de transformation linéaire (2, 2)
CD1_1	Double	Matrice d'échelle (1, 1)
CD1_2	Double	Matrice d'échelle (1, 2)
CD2_1	Double	Matrice d'échelle (2, 1)
CD2_2	Double	Matrice d'échelle (2, 2)
PLTSOLVD	Logique	Solver Siril

5.4 Astro-TIFF

En 2022, Han Kleijn, développeur du logiciel ASTAP, a proposé de contribuer au développement d'un nouveau pseudo-standard utilisant le format TIFF et tirant parti de la puissance des en-têtes de fichiers FITS. C'est ainsi qu'est né le format Astro-TIFF.

5.4.1 Pourquoi un nouveau standard parmi tous les autres ?

Actuellement, le format le plus utilisé pour l'astrophotographie est le format FITS. Celui-ci, développé par des scientifiques professionnels, répond à toutes les attentes des amateurs. Et bien que sa grande flexibilité pose quelques problèmes de compatibilité, il reste le format à privilégier.

D'autres formats spécialisés existent mais sont généralement associés à un logiciel spécifique. C'est le cas du format XISF développé par l'équipe PixInsight. Ce dernier format, bien que souvent demandé dans Siril, est un format dédié à PixInsight, un logiciel propriétaire. L'intérêt de développer une compatibilité avec Siril est donc minime et nous l'avons fait seulement en lecture pour le cyle de version 1.4.x.

Le développement d'Astro-TIFF apparaît alors comme une bonne alternative, car basé sur le format TIFF, il est possible d'ouvrir les fichiers sur n'importe quel logiciel de traitement d'images.

Enfin, le format TIFF supporte la compression (tout comme le format FITS), ce qui permet de réduire la taille des images.

5.4.2 Spécification 1.0

Daté du 2022-06-21

— Les fichiers sont conformes à la spécification TIFF 6.0, y compris le supplément 2.
— L'en-tête FITS est écrit dans la balise de base TIFF **Description de l'image**. Code 270, Hex 010E.
— L'en-tête est conforme à la spécification FITS, sauf que les lignes peuvent être inférieures à 80 caractères et que les lignes se terminent par CR+LF (0D0A) ou LF (0A).
— La première ligne de la description est la première ligne de l'en-tête et commence par `SIMPLE`. La dernière ligne de l'en-tête commence par `END`.

Recommandations

— `TIFFtag_orientation=1` (left-top) L'orientation suit les conventions. Le pixel `FITS_image[1,1]` est en bas à gauche. Le pixel `TIFF_image[0,0]` est en haut à gauche. Ces pixels sont d'abord écrits ou lus dans le fichier. Ainsi, lors de l'écriture d'une image FITS en TIFF en préservant l'orientation pour l'utilisateur, le premier pixel à écrire est `FITS_image[1,NAXIS2]`.
— `TIFFtag_compression=8` (Deflate) ou 5 (LZW).
— Pour les images en niveaux de gris, `TIFFtag_PhotometricInterpretation = 1` (la valeur minimale est le noir, la valeur maximale est le blanc).
— Indique tous les mots-clés d'en-tête disponibles.

Notes

— Cette utilisation du format TIFF est prévue pour les brutes, les darks, les flats et les darks de flat en 16 bits (images astronomiques), mais peut également être utilisée dans le format 32 bits. Il est possible de convertir FITS en TIFF et inversement, mais le programmeur de l'application peut décider d'exporter uniquement (écriture) ou d'importer uniquement (lecture) au format Astro-TIFF.
— Si une solution astrométrique est incluse, elle doit correspondre à l'orientation de l'image.
— Certains mots-clés d'en-tête sont redondants comme `NAXIS1`, `NAXIS2`, `BZERO` et `BITPIX` et ne sont pas nécessaires. Les dimensions et le type d'image TIFF sont prépondérants.
— Le modèle de dématriçage spécifié dans l'en-tête doit correspondre à l'orientation de l'image.
— L'en-tête est visible dans de nombreux programmes de manipulation d'images.

Exemple d'un en-tête Astro-TIFF qui ressemble à un en-tête de fichier FITS :

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

(suite sur la page suivante)

(suite de la page précédente)

```
RA      =              144.979 / Image center Right Ascension (deg)
DEC     =              70.0028 / Image center Declination (deg)
CRPIX1  =               3123.5 / Axis1 reference pixel
CRPIX2  =               2088.5 / Axis2 reference pixel
CRVAL1  =              144.979 / Axis1 reference value (deg)
CRVAL2  =              70.0028 / Axis2 reference value (deg)
CD1_1   =         -0.000580606 / Scale matrix (1, 1)
CD1_2   =         -4.12215e-05 / Scale matrix (1, 2)
CD2_1   =         -4.11673e-05 / Scale matrix (2, 1)
CD2_2   =          0.000580681 / Scale matrix (2, 2)
PLTSOLVD=                    T / Siril internal solve
HISTORY Background extraction (Correction: Subtraction)
HISTORY Plate Solve
END
```

5.4.3 Sauvegarder un Astro-TIFF dans Siril

Dans Siril, vous pouvez enregistrer des fichiers Astro-TIFF en choisissant le format TIFF dans la boîte de dialogue d'enregistrement lorsque vous cliquez sur *Enregistrer sous*. La liste déroulante de la boîte de dialogue TIFF vous permet de choisir entre l'enregistrement au format TIFF standard ou au format Astro-TIFF. Ce dernier est le format par défaut.

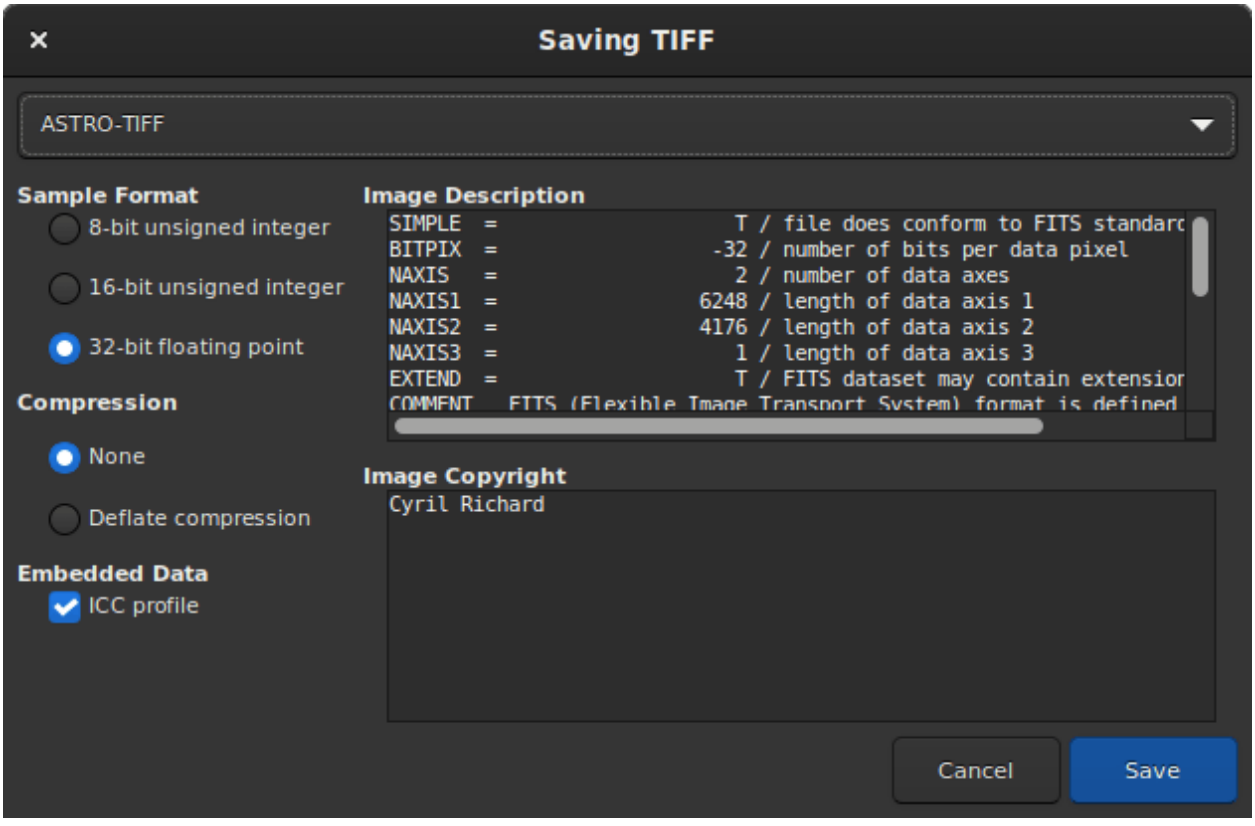


FIG. 4 – Boîte de dialogue d'enregistrement avec l'option Astro-TIFF

Ligne de commande Siril

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

Enregistre l'image actuelle sous la forme d'un fichier TIFF non compressé de 16 bits par canal : **filename**.tif. L'option **-astro** permet d'enregistrer au format Astro-TIFF, tandis que **-deflate** active la compression.

Voir aussi SAVETIF32 et SAVETIF8

5.4.4 Exemples de fichiers

— Fichier Astro-TIFF créé par Siril (32-bit, non compressé).
— Fichier Astro-TIFF créé par Siril (32-bit, compressé).

5.5 SER

Le format de fichier SER est un format de séquence d'images simple, similaire aux films non compressés. La documentation se trouve sur la page officielle. Le dernier document PDF est en miroir sur free-astro également.

Avec les améliorations des versions 2 et 3, SER gère les images en couleur, ce qui le rend parfait pour remplacer le format AVI ou d'autres formats de films produits par les anciens programmes de capture dans toutes les situations d'astronomie. Les images compressées ne doivent pas être utilisées pour l'astronomie, mais peuvent être converties en SER, ce qui rendra les fichiers plus volumineux pour la même qualité, mais plus faciles à travailler.

Siril peut convertir n'importe quelle séquence d'images et de nombreux formats de films en fichiers SER. Ser-player est un outil formidable qui permet de visualiser les fichiers SER comme n'importe quel film, avec de nombreuses options et qui fonctionne sur la plupart des systèmes d'exploitation.

Le principal problème avec AVI et d'autres conteneurs de films est qu'ils sont conçus pour fonctionner avec de nombreux codecs et formats de pixels, ce qui est une bonne chose pour les films d'usage général, mais exige que les logiciels d'astronomie gèrent un large éventail de formats de fichiers réellement différents. Les logiciels de cinéma général ne sont souvent pas bien équipés pour gérer des valeurs de 16 bits par pixel ou certains formats de données non compressées. Avec SER, un seul format de fichier permet de tout gérer, c'est pourquoi Siril, par exemple, développe maintenant un traitement uniquement pour SER.

5.5.1 Structure de fichier

Un fichier SER est composé de trois parties :
— Une en-tête de 178 bytes contenant les informations sur les images et l'observation
— les données d'image, les données de pixels brutes
— un trailer facultatif contenant les dates de toutes les images de la séquence

5.5.2 Traitement des couleurs

Dans la version 3 (2014), il y a deux façons de traiter les images colorées dans SER. Si les données proviennent directement d'un capteur, la méthode préférée est probablement d'utiliser des images à un plan et d'interpoler les données du réseau de filtres de couleur (de manière similaire aux formats de fichiers CFA utilisés dans les logiciels d'astronomie).

L'autre méthode, ajoutée dans la version 3, consiste à utiliser trois plans pour représenter les données de l'image RVB. SER v3 prend en charge les données RVB/BGR 8/16 bits. Cela peut être utile si les données sont converties à partir d'une source dont la matrice de filtres de couleur est inconnue ou pour une conversion générale.

5.5.3 Problème de spécification avec l'endianness

Étant donné que les fichiers SER peuvent contenir des images d'une précision de 16 bits, le boutisme doit être bien spécifiée. La spécification autorise le boutisme à être soit gros-boutiste, soit petit-boutiste, pour faciliter la création de fichiers sur différents systèmes, à condition que le boutisme utilisé soit documenté dans l'en-tête du fichier.

Pour une raison inconnue, plusieurs des premiers programmes à supporter SER ne respectent pas la spécification concernant l'indicateur de boutisme. La spécification indique qu'une valeur booléenne est utilisée pour l'en-tête petit-boutiste, et ils l'utilisent comme un en-tête gros boutiste, avec 0 pour petit-boutiste et 1 pour gros-boutiste. Par conséquent, pour ne pas rompre la compatibilité avec ces premières implémentations, des programmes ultérieurs, comme Siril, GoQat, Ser-player et bien d'autres, ont également décidé d'implémenter cet en-tête dans le sens opposé à la spécification.

5.6 IRIS PIC

Le format PIC est un format d'image propriétaire créer pour le logiciel IRIS de Christian Buil. Dans un soucis de compatibilité avec ce dernier, Siril est capable de lire ce type de fichiers. Cependant, du fait que le format est propriétaire et que les spécifications ne soient pas connues, toutes les informations contenues dans l'entête ne seront pas sauvegardées lors de la conversion en FITS.

Siril ne peut pas enregistrer au format PIC.

5.7 PixInsight XISF

PixInsight a introduit une structure de fichier appelée Extensible Image Serialization Format ou XISF. Ce format a été écrit pour remplacer le format FITS, couramment utilisé en astrophotographie. Cependant, ce format a été essentiellement conçu par et pour le logiciel PixInsight, et de nombreuses métadonnées importantes sont stockés dans des espaces de noms privés qui, selon l'avis des développeurs mêmes de Pixinsight, ne sont pas fiables et peuvent changer arbitrairement d'une version de spécification XISF à l'autre. De plus, la façon dont un mot clé est utilisé pour décrire telle ou telle caractéristique peut s'avérer très opaque. Par conséquent, les avantages qu'il pourrait apporter pour une utilisation dans Pixinsight sont généralement nuls pour d'autres logiciels. En outre, le format de fichier FITS est un format reconnu et largement utilisé par la communauté. La NASA elle-même l'utilise pour ses propres images. Il est très flexible et puissant, et contient toutes les fonctionnalités nécessaires au traitement astrophotographique moderne.

Nous avons donc décidé de permettre à Siril de lire ce type de fichier, mais en mode lecture seule. La sauvegarde au format XISF n'est pas et ne sera pas implémentée.

CHAPITRE 6

Gestion des couleurs

Siril offre maintenant un flux de travail de gestion de couleurs utilisant les profils ICC. En assurant une reproduction précise et cohérente des couleurs sur différents appareils, Siril vous permet d'éditer vos images en toute confiance, en sachant que les couleurs que vous voyez sur votre écran se traduiront fidèlement sur des tirages ou d'autres écrans. Vous pouvez éditer vos images dans des espaces colorimétriques mieux adaptés aux capacités d'une imprimante photo professionnelle à large gamme, permettant ainsi la production de tirages avec des couleurs plus riches que ce qui est possible en travaillant dans l'espace colorimétrique standard sRGB, et vous pouvez produire des images que votre public sera en mesure de visualiser de manière cohérente avec la façon dont vous les voyez pendant l'édition, quel que soit leur périphérique d'affichage.

Il y a plusieurs pages dans cette section de la documentation : elles visent à fournir un aperçu de ce que fait la gestion des couleurs, comment elle peut vous aider et comment elle fonctionne dans Siril.

6.1 Pour commencer

Bienvenue dans le nouveau système de gestion des couleurs de Siril ! La gestion de la couleur peut sembler complexe, c'est pourquoi cette page a pour but de fournir les bases absolues que vous devez connaître pour commencer. Elle couvre deux configurations pour vous faciliter la vie.

6.1.1 Configuration par défaut

Les paramètres par défaut offrent un comportement similaire à celui des versions précédentes de Siril.

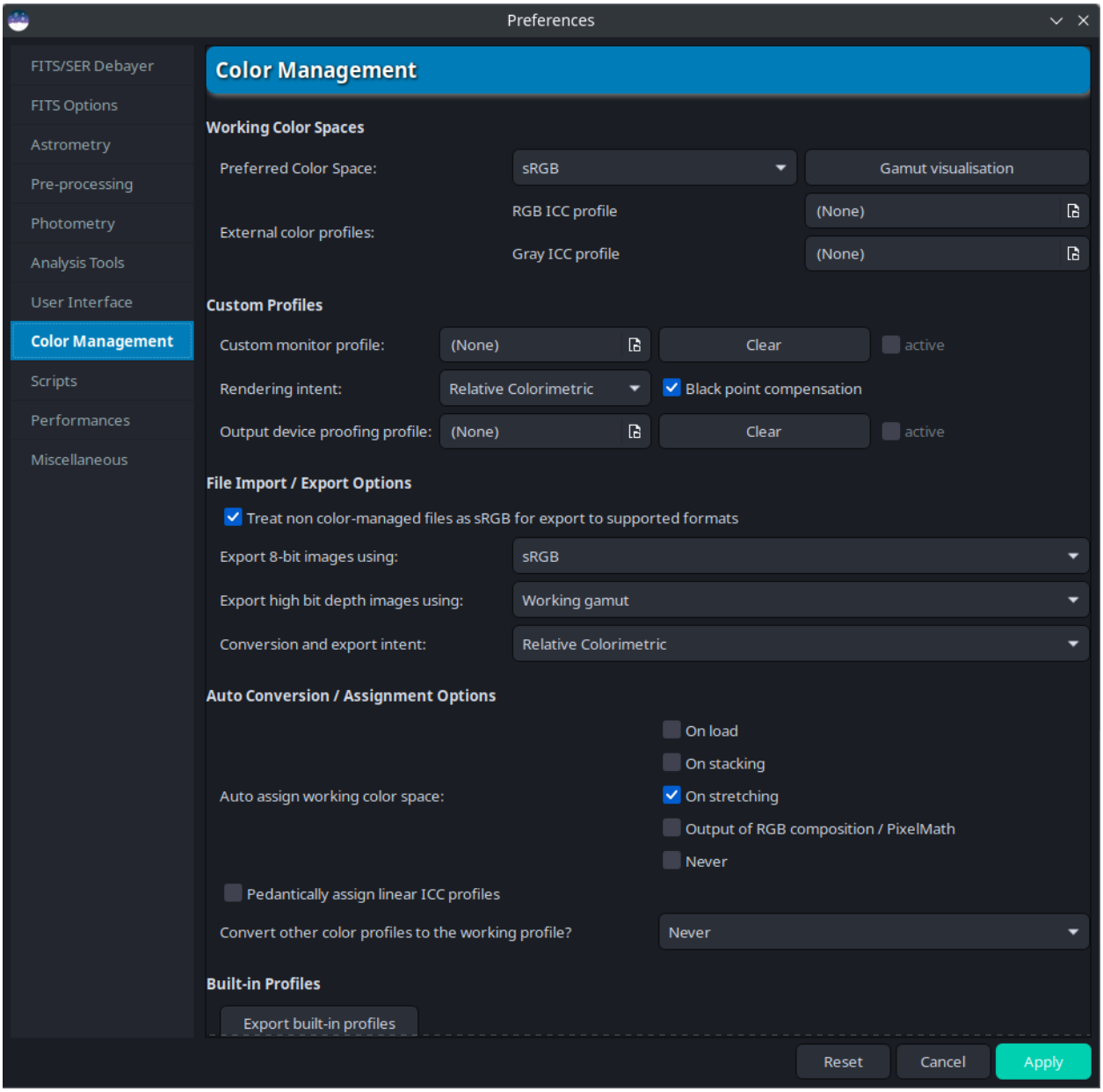
Le moniteur est défini comme sRGB et l'espace colorimétrique par défaut est également défini comme sRGB. Cela signifie qu'aucune transformation n'est nécessaire entre les deux et qu'il n'y a donc pas de surcharge de traitement pour les images utilisant le profil par défaut.

Lorsque les images sont chargées, si elles ont un profil de couleur associé, ce profil sera préservé. Cela signifie qu'une transformation d'affichage est nécessaire et entraîne donc un léger surcoût de traitement, mais minimise les changements apportés au fichier chargé.

Si les images n'ont pas de profil de couleurs, l'espace colorimétrique par défaut (sRGB) leur est attribué lors du premier étirement. C'est à ce moment-là que vous devez utiliser le mode de visualisation linéaire pour que l'image apparaisse à l'écran comme vous le souhaitez, et c'est donc à ce moment-là que la gestion des couleurs devient importante.

Les images seront exportées dans tous les formats 8 bits et 16 bits en utilisant l'espace colorimétrique sRGB.

Cette configuration par défaut convient parfaitement si votre objectif principal est la production d'images pour le web ou la visualisation sur les écrans d'ordinateur les plus courants.

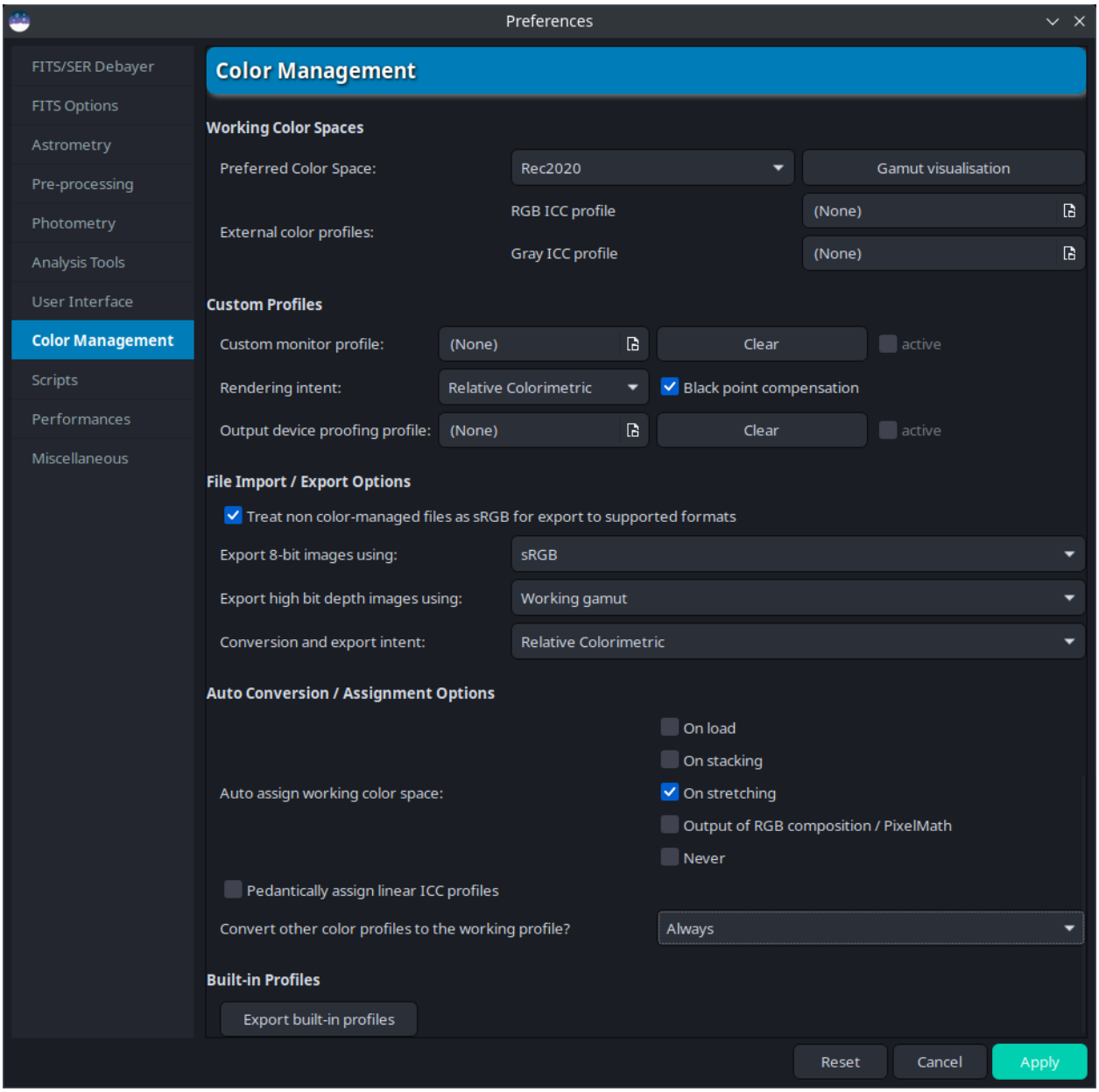


6.1.2 Configuration de l'impression photo

Si vous avez l'intention de produire des tirages photo de votre travail, envisagez la configuration suivante.

Le moniteur reste sRGB dans la capture d'écran, car aucun profil de moniteur personnalisé n'est défini, mais si vous avez accès au matériel nécessaire pour calibrer votre moniteur, vous pouvez définir un profil de moniteur personnalisé adapté à votre affichage. (La manière de procéder sort du cadre du Siril, mais vous pouvez consulter le logiciel libre ArgyllCMS.)

L'espace colorimétrique préféré est Rec2020. Cet espace colorimétrique à large gamut est mieux adapté à un flux de travail destiné à l'impression, car la conversion RVB-CMJN effectuée par l'imprimeur est moins susceptible d'altérer le


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

profil CMJN.

Lorsque les images sont chargées, si elles ont un profil de couleur associé, elles sont converties en Rec2020. Si les images n'ont pas de profil de couleur, le profil préféré (Rec2020) leur est attribué lors du premier étirement.

6.1.3 Intention du rendu

Dans les deux configurations proposées, l'intention du rendu est réglée sur Colorimétrie relative. C'est ce que vous devriez toujours utiliser pour éditer votre image, car avec ceci, les couleurs dans le gamut restent vraies et cela donne la vision la plus précise de ce à quoi ressemble votre image dans son ensemble.

Cependant, l'intention Colorimétrique Relative permet d'écrêter les couleurs hors gamut. Vous ne verrez pas les détails de la coloration qui sont hors gamut.

Il peut donc arriver que vous souhaitiez changer le mode de rendu en mode perceptuel. Les couleurs seront alors transférées en douceur de l'espace de travail à l'écran. Tous les détails de la coloration sont affichés, mais au prix d'une perte de saturation globale. Cette perte est inévitable, car le gamut plus large est écrasé pour s'adapter au gamut plus petit. Par conséquent, le mode perceptuel ne fournit pas une véritable visualisation de votre image : il ne devrait **pas** être utilisé pour l'édition, **uniquement** pour explorer les détails de vos images qui se situent en dehors du gamut sRGB. Voir la discussion ici.

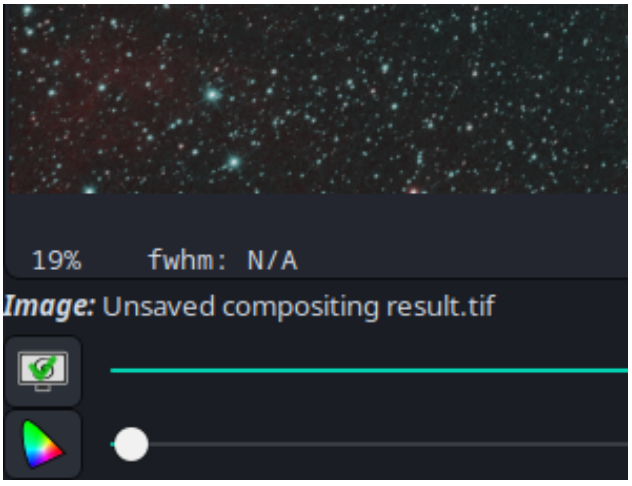
Les images seront exportées vers des formats 8 bits comme sRGB, mais vers des formats à grande profondeur de bits comme Rec2020 (et vous devriez enregistrer votre version finale dans un format 16 bits standard de l'industrie de l'imprimerie, c'est-à-dire probablement TIFF).

6.2 État de la gestion des couleurs

L'état de la gestion des couleurs de l'image chargée est indiqué par l'icône d'état de la gestion des couleurs dans le coin inférieur gauche de la fenêtre. L'icône est entièrement saturée lorsque l'image est gérée en couleur, et désaturée lorsque l'image n'est pas gérée en couleur. L'infobulle fournit des informations d'état plus détaillées si l'image est gérée en couleur : elle montre l'espace de travail et les profils de moniteur.

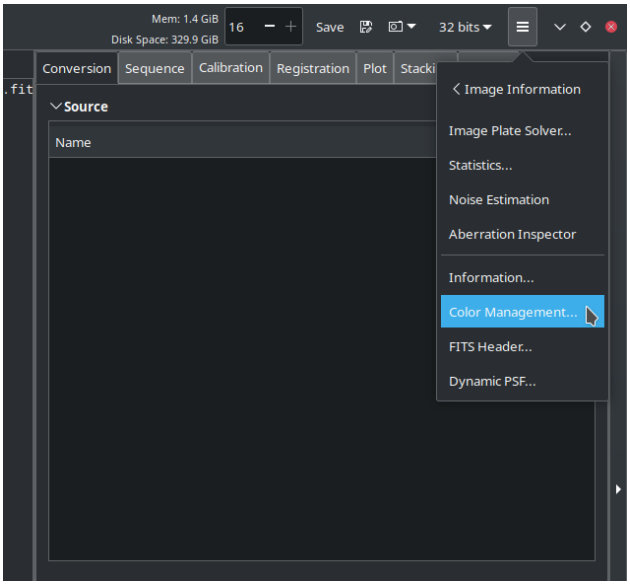
Un clic gauche sur le bouton d'état permet d'accéder à la boîte de dialogue de gestion des couleurs. Un clic droit sur l'outil permet de basculer vers le mode d'affichage de l'évaluation des couleurs ISO 12646. Ces deux modes sont décrits plus en détail ci-dessous.

Juste au-dessus du bouton d'état de la gestion des couleurs se trouve un nouveau menu pour les contrôles d'image. Il inclut le contrôle de coupe existant, qui met en évidence les pixels qui dépassent le curseur "haut" de l'interface utilisateur, et ajoute un nouveau contrôle de gamut, qui met en évidence les pixels qui sont hors gamut pour le profil d'épreuvage doux en magenta vif.



6.3 Boîte de dialogue de gestion des couleurs

L'outil de gestion des couleurs est accessible en cliquant sur l'icône d'état Gestion des couleurs. (Il est également possible d'y accéder par le menu *Outils*.)



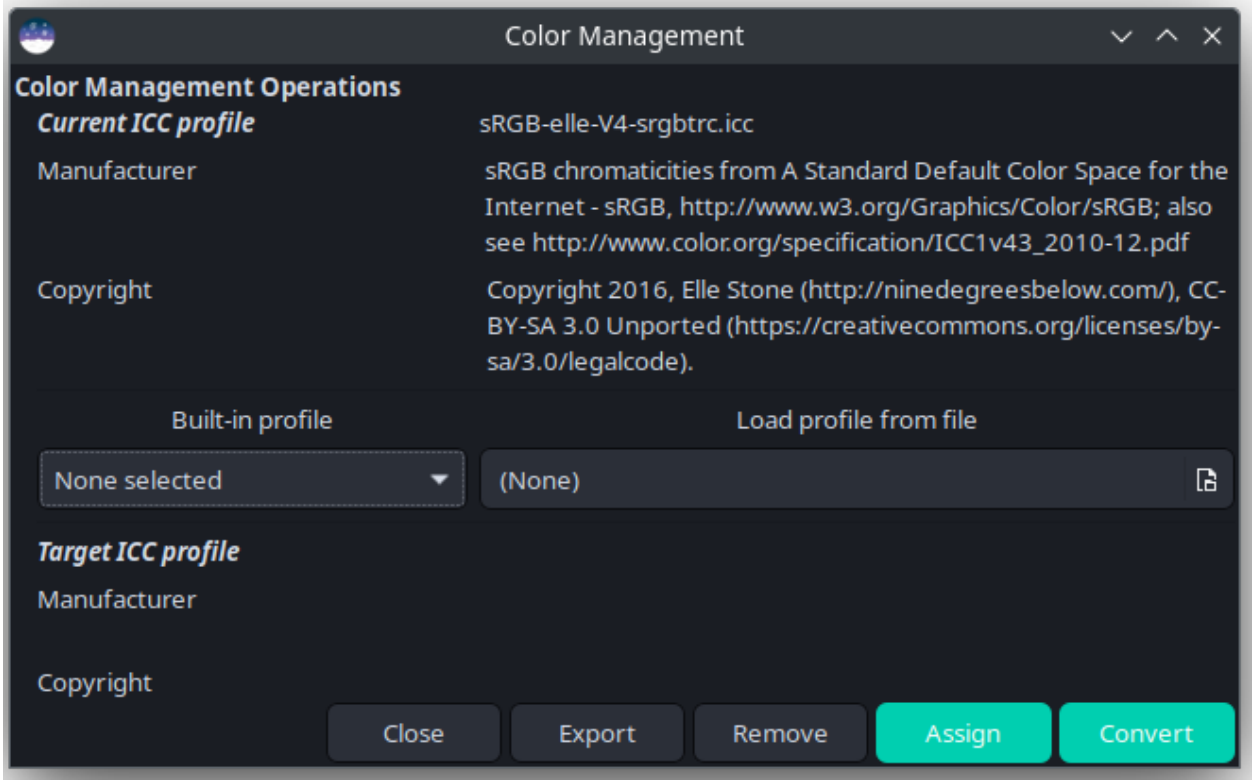
L'outil lui-même est présenté ci-dessous.

La partie supérieure de l'outil affiche des informations sur le profil ICC actuel attribué à l'image. La description, les détails du fabricant et l'avis de copyright sont affichés.

En dessous, des sélecteurs vous permettent de choisir un profil différent, soit l'un des profils intégrés à l'aide du menu déroulant de gauche, soit un profil ICC que vous souhaitez charger à partir d'un fichier à l'aide du sélecteur de fichiers de droite.

Avertissement

Lors du chargement d'un profil à partir d'un fichier, le type de profil doit être soit RVB soit Gris pour correspondre à l'image chargée. L'affectation ou la conversion d'images à d'autres espaces colorimétriques tels que XYZ, CIE La*b* ou CMYK n'est pas prise en charge. Toutefois, des espaces colorimétriques tels que CIE La*b* peuvent être utilisés en interne par certaines opérations d'image, le cas échéant.



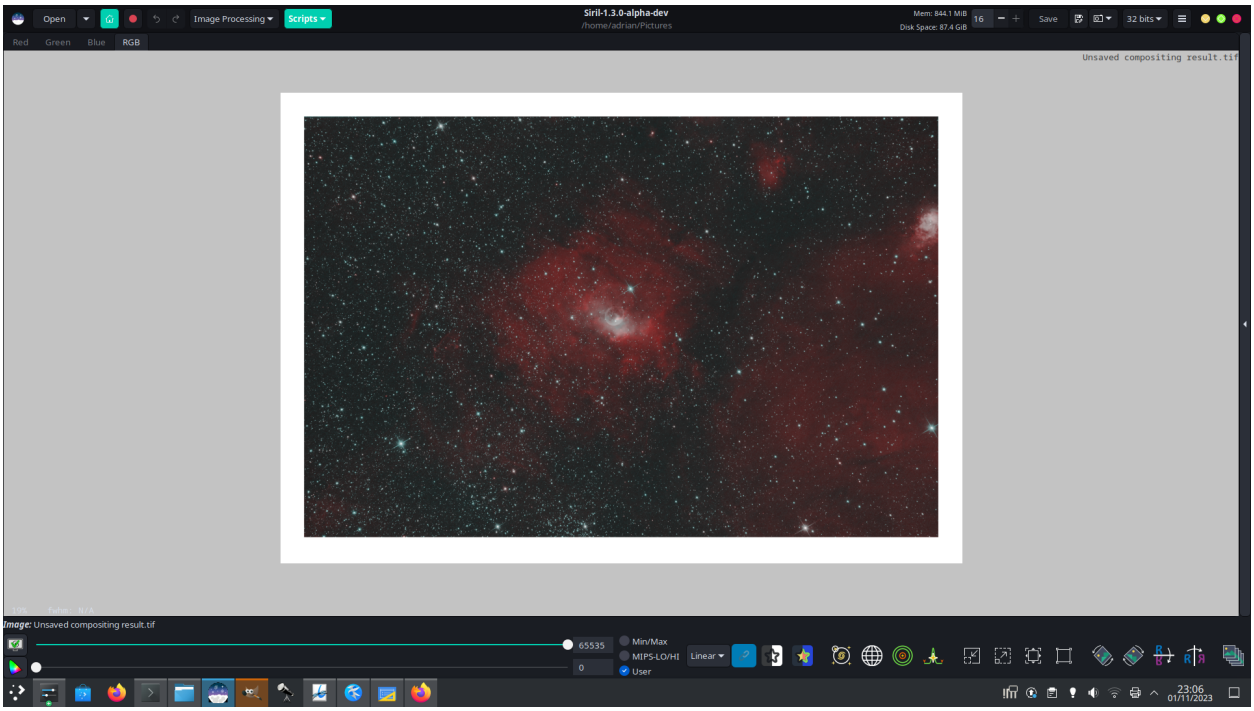
Astuce

Vous pouvez convertir une image RVB en un profil Gris. Il en résultera une image mono avec la transformation de la courbe de réponse tonale (CRT) appropriée et les contributions des 3 canaux R, V et B déterminées par les points blancs des deux profils. La conversion d'une image grise en profil RVB donnera une image à 3 canaux avec la transformation CRT appropriée et toutes les composantes des pixels égales.

En bas de la fenêtre, l'outil affiche le même type d'informations pour le profil cible que pour le profil original en haut de la fenêtre. Cela permet de vérifier que vous avez chargé le bon profil.

6.3.1 Barre d'outils

- Le bouton *Exporter* exporte simplement le profil de l'image courante dans un fichier du répertoire de travail.
- Le bouton *Supprimer* supprime le profil ICC d'une image.
- Le bouton *Assigner* assigne le profil ICC sélectionné à l'image, sans effectuer de conversion d'espace couleur. Ceci est utile si l'image contient un profil de couleur incorrect.
- Le bouton *Convertir* convertit l'image au profil ICC sélectionné et assigne le profil à l'image. Ceci est utile si vous souhaitez transformer l'image de son espace colorimétrique actuel vers un autre espace colorimétrique.



Avertissement

L'utilisation des boutons *Supprimer*, *Assigner* ou *Convertir* (ou des commandes Siril *icc_remove*, *icc_assign* ou *icc_convert_to*) efface toute ROI définie.

6.4 Mode d'évaluation des images ISO 12646

La norme ISO 12646 définit les conditions d'observation optimales pour l'évaluation des couleurs. En résumé, pour obtenir la meilleure évaluation des couleurs de votre image, les meilleures conditions d'observation sont fournies par un fond gris neutre uniforme appelé D50. Techniquement, cela devrait même s'étendre aux murs de votre pièce, mais Siril ne peut pas contrôler cela ! Siril propose cependant un mode d'évaluation de l'image qui se rapproche des recommandations de la norme ISO 12646. Il est disponible par un clic droit sur le bouton de gestion des couleurs en bas à gauche de l'écran, et ne nécessite pas que l'image soit gérée en couleur pour l'utiliser.

Ce mode est destiné à la vérification finale des couleurs de l'image. Il règle donc le mode de prévisualisation sur linéaire avec les curseurs à 0 et 65535 (dynamique complète). Il masque également le panneau et règle le zoom de manière à ce que l'image complète soit visible et centrée avec un large bord autour d'elle, et définit un bord blanc modéré autour de l'image pour fournir une référence visuelle blanche, entourée d'un arrière-plan de gris D50.

Idéalement, vous devriez également choisir un thème GTK gris neutre, tel que l'excellent thème Equilux.

Ce mode se désactive automatiquement si le zoom est modifié ou si le panneau est à nouveau visible. Il peut être désactivé en cliquant une seconde fois avec le bouton droit de la souris sur le bouton de gestion des couleurs. La vue reprendra ses paramètres de zoom et son état de panneau précédents.

6.5 Commandes

La plupart des commandes Siril n'interviennent pas dans la gestion des couleurs, sauf qu'elles n'empêchent pas la préservation des profils de couleurs existants dans une image.

Les exceptions sont trois commandes spécifiques de gestion des couleurs qui peuvent être utilisées pour attribuer, convertir ou supprimer le profil ICC dans une image.

Ligne de commande Siril

```
icc_assign profile
```

Affecte le profil ICC spécifié dans l'argument à l'image actuelle.
L'un des arguments spéciaux suivants peut être fourni pour utiliser les profils intégrés respectifs : **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **working** pour définir le profil de couleur de travail monochrome ou RVB, (pour les images monochromes uniquement) **linear**, ou le chemin d'accès à un fichier de profil ICC peut être fourni. Si un profil intégré est spécifié avec une image monochrome chargée, le profil Gray avec la CRT correspondant sera utilisé

Ligne de commande Siril

```
icc_convert_to profile [intent]
```

Convertit l'image actuelle au profil ICC spécifié.
L'un des arguments spéciaux suivants peut être fourni pour utiliser les profils intégrés respectifs : **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **graysrgb**, **grayrec2020**, **graylinear** ou **working** pour définir le profil de travail de couleur monochrome ou RVB, (pour les images monochromes uniquement) **linear**, ou le chemin d'accès à un fichier de profil ICC peut être fourni. Si un profil intégré est spécifié avec une image monochrome chargée, le profil Gray avec la CRT correspondant sera utilisé.

Un deuxième argument peut être fourni pour spécifier l'intention de transformation de la couleur : il doit s'agir d'un des éléments suivants : **perceptual**, **relative** (pour la colorimétrie relative), **saturation** ou **absolute** (pour la colorimétrie absolue)

Ligne de commande Siril

```
icc_remove
```

Supprime le profil ICC de l'image actuelle, si elle en possède un

6.6 Matrices de conversion des couleurs

6.6.1 Introduction

Les profils ICC ne couvrent pas tout ce qu'un utilisateur peut souhaiter faire en termes de manipulation des couleurs, c'est pourquoi des outils supplémentaires sont également fournis. Bien sûr,*PixelMath* est un outil général puissant pour manipuler les pixels, mais un cas d'utilisation courant est l'application d'une matrice de correction des couleurs aux données, par exemple pour la conversion manuelle des chromatismes de l'appareil photo.

Avertissement

L'application des matrices de conversion des couleurs (CCM) est une technique avancée. La description des techniques impliquant l'utilisation des CCM sort du cadre de la documentation Siril. Pour utiliser cet outil avec succès, vous devez comprendre le fonctionnement des CCM et la manière de les appliquer dans votre flux de travail.

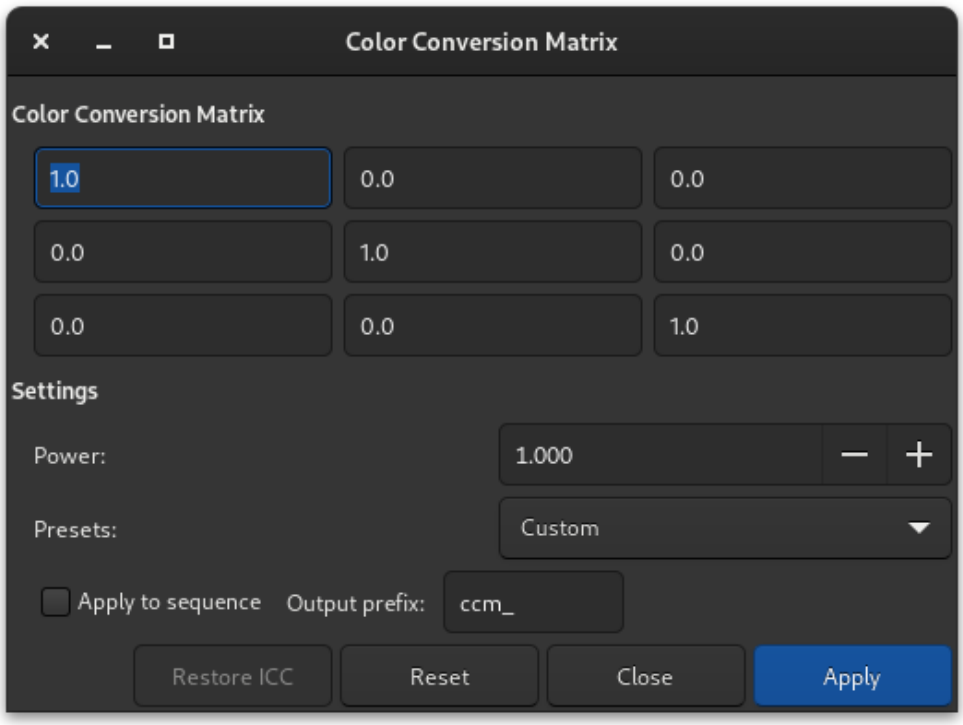


FIG. 1 – Fenêtre des matrices de conversion des couleurs.

6.6.2 Outil de matrice de conversion des couleurs

L'outil*Outils* → *Matrice de conversion des couleurs* permet d'appliquer directement une matrice de conversion des couleurs (CCM) aux pixels. La CCM est spécifiée par 9 éléments :

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

Plusieurs préréglages courants sont proposés dans un menu déroulant. L'outil offre en outre la possibilité de mettre l'image à l'échelle à l'aide facteurγ.

Elle s'applique aux pixels de la manière suivante :

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

Avertissement

Si un CCM est appliqué à une image qui a un profil ICC intégré, le profil ICC ne sera plus une description valide des données de l'image. Le profil est donc temporairement désactivé et l'icône de gestion des couleurs apparaît comme inactive. On suppose que votre flux de travail implique des transformations d'espace colorimétrique de bas niveau et des opérations sur l'image et qu'à un moment donné, vous finirez par retransformer les données dans l'espace colorimétrique décrit par le profil ICC. À ce stade, le profil ICC peut être réactivé en utilisant la partie inférieure de la boîte de dialogue. Si toutefois votre flux de travail implique une conversion manuelle de l'image vers un espace colorimétrique final différent, vous devrez appliquer le profil ICC cible en utilisant la boîte de dialogue*Gestion des couleurs*.

Notez que ceci ne s'applique pas à la commande*ccm* en ligne de commande. Par principe, les commandes Siril n'interagissent pas avec les profils ICC, donc la commande*ccm* ne désactivera pas un profil ICC attaché à une image : il est de votre responsabilité de le faire en utilisant la commande*icc_remove* si nécessaire.

Cette opération peut être appliquée à des séquences. Ouvrez une séquence et préparez les paramètres que vous voulez utiliser, puis cochez le bouton*Appliquer à la séquence* et définissez le préfixe de sortie de la nouvelle séquence (`ccm_` par défaut).

Ligne de commande Siril

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

Applique une matrice de conversion des couleurs à l'image en cours.

Il y a 9 arguments obligatoires correspondant aux 9 éléments de la matrice :

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

Un dixième argument supplémentaire **[gamma]** peut être fourni : s'il est omis, la valeur par défaut est 1,0.

Ceux-ci sont appliqués à chaque pixel selon les formules suivantes :

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

Ligne de commande Siril

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

Même commande que CCM mais pour la séquence **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

Le nom de la séquence de sortie commence par le préfixe "ccm_", sauf indication contraire avec l'option **-prefix=**

Liens :*ccm*

6.7 Théorie de la gestion des couleurs

6.7.1 Introduction

La gestion des couleurs est la méthode utilisée pour s'assurer que les couleurs d'une image sont toujours cohérentes, quelle que soit la manière dont l'image est visualisée. Pour ce faire, on utilise des profils de couleurs (profils ICC, du nom de l'International Color Consortium). Chaque type d'écran et chaque combinaison imprimante/papier possède son propre profil de couleurs. L'image a également un espace colorimétrique défini et, en transformant l'image entre différents espaces colorimétriques lorsqu'elle est visualisée sur deux types d'écrans différents ou lorsqu'elle est imprimée, nous nous assurons qu'elle a toujours la même apparence (ou du moins qu'elle s'en rapproche le plus possible).

6.7.2 Espace colorimétriqueCIE 1931

L'espace colorimétrique CIE 1931 (CIE 1931) cartographie toutes les couleurs perceptibles par l'œil humain. Comme l'œil humain possède trois types de cellules coniques (récepteurs de couleurs), la CIE 1931 a trois paramètres (X, Y et Z). Notez que les paramètres X, Y et Z ne correspondent pas directement à la réponse de chaque type de cellule conique, mais ils permettent trois degrés de liberté. Pour plus de détails sur cet espace colorimétrique, voir ici : https://en.wikipedia.org/wiki/CIE_1931_color_space

Il ne s'agit pas d'un espace couleur qui est utilisé pour stocker des données d'image la plupart du temps, mais il est important parce qu'il est utilisé comme espace couleur intermédiaire et parce qu'il définit les "vraies" couleurs (celles qui sont visibles par l'œil humain) et les "imaginaires" (celles que l'on ne peut pas voir). La CIE 1931 peut être visualisée sous la forme d'un fer à cheval.

Notez que l'extérieur du fer à cheval définit la couleur des lignes spectrales monochromatiques pures. Ce point sera important par la suite.

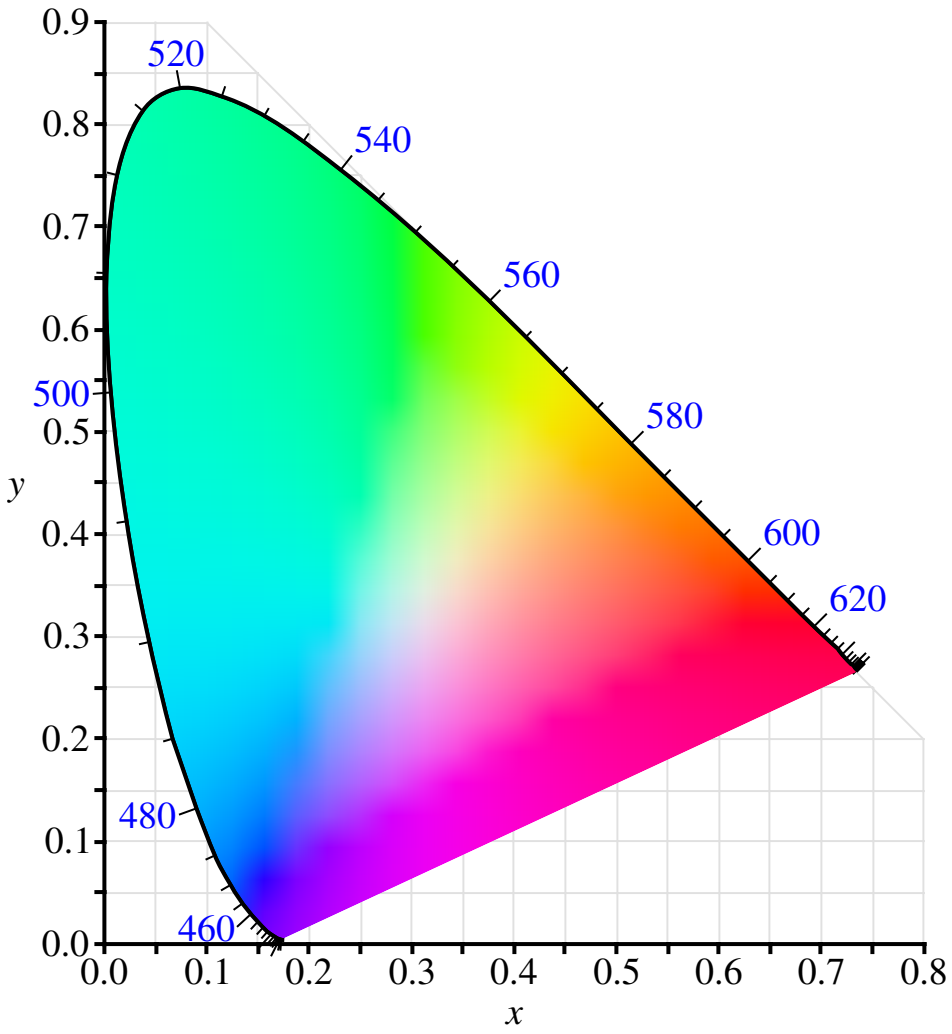


FIG. 2 – Par BenRG - CIExy1931.svg, Domaine public, https://commons.wikimedia.org/w/index.php?curid=7889658

6.7.3 Espaces colorimétriques RGB

Nos espaces colorimétriques de travail sont généralement basés sur les axes de couleur rouge, vert et bleu. Cela correspond à peu près au fonctionnement de nos yeux ainsi qu'aux phosphores des émetteurs d'un écran. Il existe cependant un grand nombre d'espaces colorimétriques RVB, chacun convenant à des usages différents et présentant des avantages et des inconvénients. Vous pouvez voir la variété des espaces colorimétriques dans ce diagramme, qui les définit par rapport à la CIE 1931.

Les canaux R, V et B d'une image sont définis en termes d'espace colorimétrique formé par trois couleurs primaires situées dans cet espace CIE 1931, plus un point blanc, plus une courbe de réponse tonale (CRT) pour chaque canal qui définit le gamma de chaque canal. *(En fait, les CRT peuvent être plus complexes qu'une simple courbe gamma, mais la plupart sont au moins approximativement une courbe gamma.)* Le diagramme ci-dessus montre que certains espaces colorimétriques sont considérablement plus grands que d'autres : c'est-à-dire qu'ils peuvent représenter une plus grande partie des couleurs visibles.

6.7.4 Espaces de couleur de gris

En astrophotographie, nous avons souvent affaire à des images monochromes. Celles-ci ne peuvent manifestement pas avoir un profil de couleur RVB. Au lieu de cela, elles reçoivent un profil de couleur de Gris. Celui-ci définit la courbe de réponse tonale (CRT) de l'image exactement de la même manière que les profils de couleur RVB définissent la CRT de chaque canal. Ainsi, quel que soit le type d'écran ou d'imprimante utilisé, les images monochromes peuvent toujours être reproduites fidèlement et avoir le même aspect (ou un aspect aussi proche que possible) sur différents supports de sortie.

6.7.5 Espace couleur de l'affichage

L'espace colorimétrique "standard" utilisé pour les images informatiques est traditionnellement le sRGB. Il a été créé par HP et Microsoft en 1996, puis normalisé sous la forme de la norme IEC 6166-2-1 :1999. Cet espace colorimétrique codifie le gamut des couleurs réalisables par les écrans de l'époque. Comme le montre la figure, il ne peut représenter qu'une petite partie de l'espace colorimétrique visible total défini par la CIE 1931. Cependant, de nombreux moniteurs actuels ne peuvent guère faire mieux que sRGB et c'est l'espace colorimétrique standard actuel pour le World Wide Web (WWW). Par conséquent, même si vous ne souhaitez pas l'utiliser à d'autres fins, il est nécessaire de l'utiliser comme profil d'exportation pour toute image que vous souhaitez afficher de manière cohérente dans les navigateurs web. C'est également l'espace colorimétrique supposé pour tout format d'image qui ne prend pas en charge les profils de couleur, et pour toute application qui ne dispose pas de gestion des couleurs.

Cependant, il est évident qu'il existe d'autres espaces de couleurs RVB avec des gamuts beaucoup plus larges, par exemple Adobe RVB, Adobe ProPhoto, Rec2020. Ces espaces peuvent représenter une part beaucoup plus importante des couleurs réelles définies dans la CIE1931. Ils peuvent également représenter un gamut de couleurs beaucoup plus large que celui de la plupart des moniteurs. Pourquoi ces couleurs sont-elles intéressantes ?

Tout d'abord, les moniteurs s'améliorent. Les moniteurs à large gamut ne sont pas encore très répandus, mais ils sont de moins en moins rares et de plus en plus abordables. Si vous éditez et visualisez vos images sur un moniteur à large gamme, vous bénéficierez de couleurs plus riches en utilisant un espace colorimétrique avec une gamme plus large. Certains téléphones modernes peuvent afficher l'ensemble du gamut P3, qui est considérablement plus grand que le sRGB et permet d'afficher une gamme de couleurs plus riche.

6.7.6 Profils d'impression

Les profils d'impression peuvent aller au-delà de sRGB dans certains domaines (et ne pas couvrir tout sRGB dans d'autres domaines). En éditant dans un espace colorimétrique à large gamme, cette richesse des couleurs - même si elle ne peut pas être montrée dans la sortie transformée en couleurs sur votre écran - est toujours présente et sera évidente dans des impressions de bonne qualité.

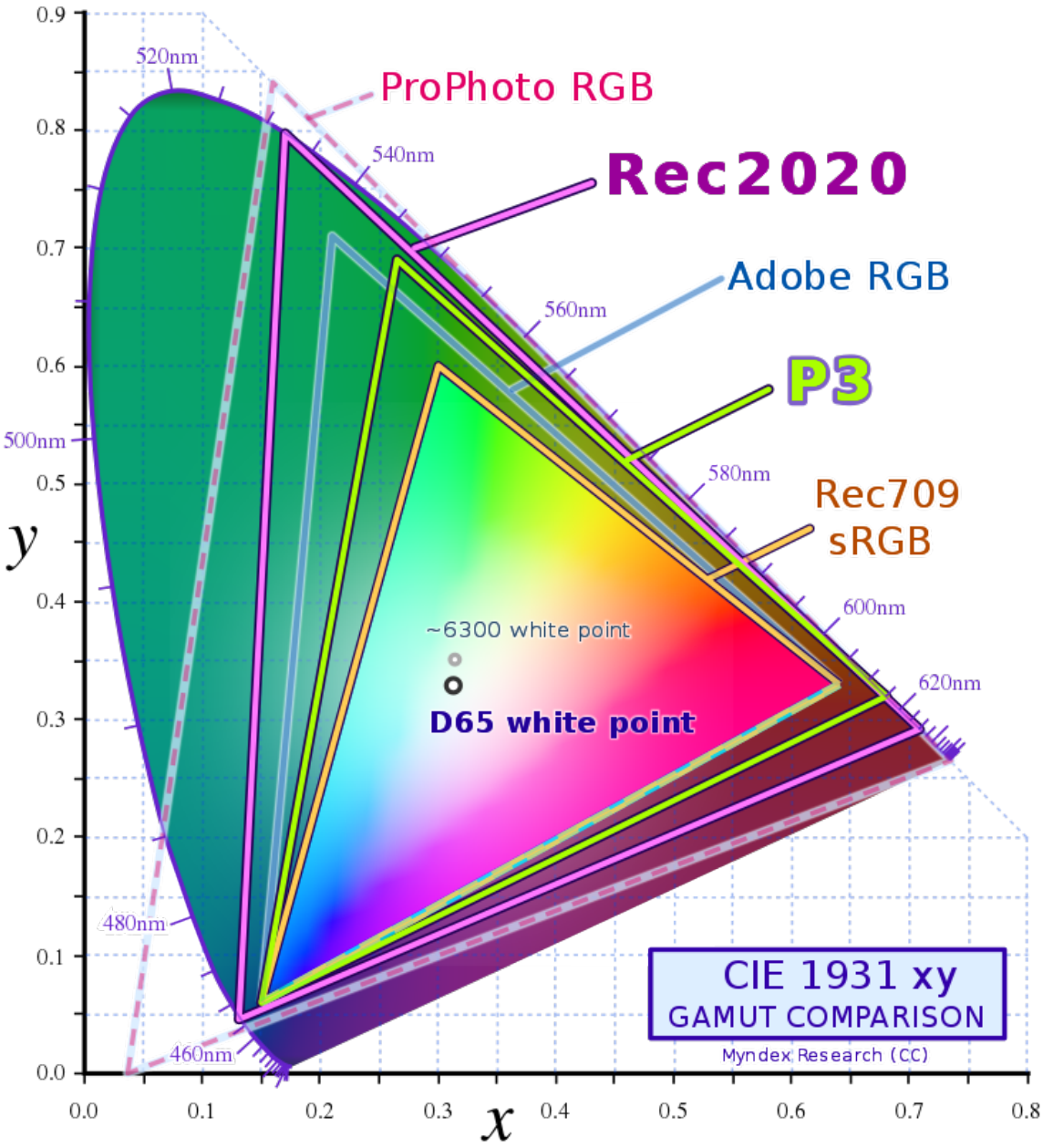


Fig. 3 – Par Myndex - Travail personnel, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=116654642

6.7.7 Transformations

L'intérêt des espaces colorimétriques est qu'une image apparaîtra souvent dans des sorties avec différents espaces colorimétriques à différents moments de sa vie. Elle peut être créée sur un moniteur à gamut élevé de qualité professionnelle, elle peut être visualisée par le public sur des moniteurs sRGB de base et elle peut être imprimée sur toute une série d'imprimantes différentes. Chacun de ces appareils est capable de reproduire des gamuts de couleurs différents, mais nous voulons que l'image, dans la mesure du possible, soit cohérente sur tous ces appareils. Pour ce faire, nous utilisons des transformations d'espace colorimétrique. Malheureusement, comme les espaces colorimétriques sont différents, nous devons résoudre le problème de la représentation, dans un espace colorimétrique, des couleurs qui sont "hors gamut" après avoir été transformées à partir d'un espace colorimétrique différent.

6.7.8 Intentions de transformation des couleurs

La réponse au problème ci-dessus est "intentions". Chaque fois que vous visualisez une image sur un appareil ou un papier dont l'espace colorimétrique est différent de votre profil de travail, une transformation de l'espace colorimétrique lui est appliquée. Ce n'est pas aussi simple qu'une correspondance arbitraire entre deux ensembles de coordonnées. Supposons que vous travailliez en Rec2020. Considérons la transformation du profil de couleur de votre moniteur. Rappelez-vous que votre moniteur (supposons qu'il s'agisse approximativement d'un écran sRGB) ne peut pas afficher autant de couleurs que celles qui sont représentables dans Rec2020. L'espace colorimétrique doit donc faire correspondre toutes les couleurs de la norme Rec2020 aux couleurs de la norme sRGB. La manière dont il le fait est déterminée par l'intention de rendu.

Vous pouvez choisir une intention pour le rendu de votre image pour l'affichage et une intention différente pour d'autres usages. Les différentes intentions définies par l'ICC et disponibles dans Siril sont décrites ci-dessous.

Astuce

- **Perceptuelle** L'intention perceptuelle met à l'échelle le gamut de couleurs d'entrée dans le gamut de couleurs de sortie. Toutes les couleurs sont modifiées, mais les relations entre les couleurs sont maintenues. En général, la couleur est moins saturée dans l'espace colorimétrique de sortie, mais la saturation par rapport aux autres couleurs est maintenue.
- **Saturation** L'intention de saturation met également à l'échelle le gamut de couleurs d'entrée dans le gamut de sortie, mais elle le fait d'une manière qui donne la priorité à la saturation. Elle convient généralement mieux aux images "éclatantes" qu'à la photographie.
- **Colorimétrie relative** L'intention colorimétrique relative reproduit fidèlement les couleurs dans le gamut, mais clippe les couleurs hors gamut sur le point le plus proche du triangle représentant les limites du profil de couleur cible.
- **Colorimétrie absolue** L'intention colorimétrique absolue n'est vraiment utile que dans le cadre de l'épreuvage avant impression.

6.7.9 Profils de couleur ICC et disponibilité des intentions

L'ICC définit les quatre intentions énumérées ci-dessus (ainsi que d'autres utilisées principalement pour le contrôle de l'encre dans les applications d'impression), mais tous les profils ICC ne prennent pas en charge toutes les intentions. La plupart des profils de couleur intégrés dans Siril sont des profils de mise en forme de la matrice. Ils sont très bons comme profils d'espace colorimétrique de travail, mais la plupart du temps, ils ne prennent en charge que l'intention colorimétrique relative. Ce n'est pas grave, car c'est la plupart du temps l'intention que nous voulons utiliser lors de la conversion entre les espaces colorimétriques. (Si vous définissez dans les préférences une intention que votre profil ICC ne supporte pas, Siril se rabattra sur une intention qui est supportée - généralement Colorimétrie Relative.

6.7.10 Affichage

Pour l'affichage, il est généralement préférable d'utiliser la méthode de colorimétrie relative. L'affichage de l'image est ainsi aussi cohérent que possible avec ce que verront les autres personnes disposant d'un écran à gestion des couleurs, ou avec ce à quoi l'image ressemblera lorsqu'elle sera imprimée dans un flux de production de gestion des couleurs.

Il se peut que vous souhaitiez parfois passer à l'option Perceptuelle. Vous n'obtiendrez pas une représentation précise des couleurs, mais vous verrez les différences relatives de couleur qui sont supprimées lorsque l'image est affichée sur votre écran. Vous pouvez vérifier quelles parties de l'image sont en dehors de votre gamut d'affichage en utilisant l'outil de vérification du gamut dans le menu de vérification de l'image - les pixels qui sont en dehors du gamut d'affichage sont affichés en magenta vif. Siril fournit un profil sRGB intégré prenant en charge l'intention perceptuelle et l'utilisera automatiquement si l'intention perceptuelle est sélectionnée dans l'onglet Préférences (et si aucun profil de moniteur personnalisé n'est actif). Si vous souhaitez utiliser un profil de moniteur personnalisé et l'intention perceptuelle, c'est à vous de vous assurer que votre profil le supporte.

6.7.11 Conversion et sauvegarde

Pour convertir les fichiers et les enregistrer, vous devez presque toujours utiliser la méthode de colorimétrie relative. Cette méthode préserve correctement les couleurs. La Colorimétrie relative coupe typiquement les couleurs dans le gamut de l'espace couleur vers lequel il est converti, mais vous obtenez un résultat cohérent. En fait, lorsqu'il s'agit d'images en 32 bits à virgule flottante, Siril utilise une transformation non bornée, c'est-à-dire qu'au lieu d'écrêter les couleurs, il autorise les valeurs négatives. Elles auront besoin d'être écrêtées à un moment donné - lors de l'affichage, par exemple - mais elles peuvent être sauvegardées, et l'application d'une transformation de couleur dans le sens inverse permet de retrouver les données d'origine. Rien n'est perdu.

6.7.12 L'épreuvage en douceur

Pour l'épreuvage à l'écran, il est préférable d'utiliser la colorimétrie relative. La colorimétrie absolue peut être utile pour simuler exactement l'aspect de votre image sur un support imprimé donné, car elle simule le point blanc d'un support sur un autre (ainsi, si vous testez l'aspect d'une image sur le blanc jaunâtre et terne du papier journal, la colorimétrie absolue tentera de simuler cet aspect sur votre moniteur), mais pour l'épreuvage à froid d'un espace colorimétrique à gamut plus large par rapport à votre profil d'affichage, vous aurez probablement besoin de la colorimétrie relative.

6.8 Flux de travail

6.8.1 Etape linéaire

Presque tous les flux de travail astrophotographiques commencent par une étape linéaire. En effet, nos capteurs photographiques sont des compteurs de photons et fournissent essentiellement une réponse linéaire aux photons entrants. Doublez le nombre de photons entrants lors d'une exposition et la valeur des pixels double (en négligeant le biais du capteur, qui sera calibré).

De nombreux flux de travail photographiques standard passent directement aux espaces colorimétriques avec une "courbe de réponse tonale" non linéaire, par exemple la CRT de sRGB est approximativement une fonction gamma $f(x) = x^{1/g}$, où $g = 2.2$.

Cependant, l'astrophotographie est différente. Représenter ces données linéaires sous forme de valeurs linéaires dans un espace colorimétrique linéaire est vital pour plusieurs raisons. Plus important encore, certaines fonctions reposent sur la linéarité des données : la détection d'étoiles correspond aux profils d'étoiles gaussiens, et si les données ont fait l'objet d'une CRT non linéaire, les profils ne seront plus gaussiens et l'algorithme de recherche d'étoiles donnera de moins bons résultats. StarNet a été formé sur des données linéaires avec une transformation spécifique (étirement de l'histogramme avec des paramètres automatiques) et encore une fois, lui fournir des données auxquelles une fonction gamma a déjà été appliquée aggravera les résultats. Déconvolution, réduction du bruit… De nombreuses fonctions que nous appliquons à nos données reposent sur leur linéarité.

Si vous le souhaitez, vous pouvez appliquer un profil linéaire à ce stade. Siril peut le faire automatiquement pour vous lorsqu'une image est ouverte, si vous définissez les préférences d'attribution automatique et cochez le bouton*Attribuer de manière rigoureuse des profils ICC linéaires*. Pour les images monochromes, le profil gris linéaire sera appliqué. Pour les images RVB (ou lors de la composition d'images monochromes linéaires en une image couleur), le profil sélectionné sera le profil linéaire de l'ensemble préféré de profils d'espace colorimétrique que vous avez défini dans*Préférences*.

Par exemple, si vous avez choisi l'espace colorimétrique Rec2020, Siril attribuera le profil ICC Rec2020-V4-g10 à votre image. « Rec2020 » est le nom du gamut ; « V4 » indique qu'il s'agit d'un profil ICC version 4 ; et "g10" indique que le profil a une CRT qui est une fonction gamma avec $g = 1.0$ (c'est-à-dire une réponse linéaire).

Astuce

L'attribution d'un profil d'espace de travail de base est suffisante pour la plupart des utilisateurs. La balance des couleurs peut être réglée ultérieurement à l'aide des outils d'étalonnage des couleurs. Cependant, si vous disposez d'une caméra calibrée, vous souhaiterez peut-être appliquer un profil d'entrée. L'étalonnage de l'appareil photo et la création de profils d'appareil photo sont un sujet avancé et ne relèvent pas de Siril, mais vous pouvez trouver des conseils dans les articles de la section D de cette page d'utilisation. La manière la plus simple d'appliquer un profil d'entrée consiste à utiliser la boîte de dialogue Gestion des couleurs. Sélectionnez votre profil de caméra dans le sélecteur de fichiers et cliquez sur *Assigner*. C'est en fait suffisant : Siril se fera un plaisir de modifier votre image dans le profil de l'appareil photo, mais vous pouvez également sélectionner votre profil de travail et cliquer sur *Convertir* pour convertir les données dans votre espace colorimétrique préféré.

Une autre approche pour calibrer votre entrée consiste à utiliser des matrices de conversion de couleurs (CCM). Cela demande plus de travail : comme pour les profils de caméra, les CCM doivent être produits en dehors de Siril, et ils peuvent ensuite être appliqués à vos données linéaires à l'aide de PixelMath.

Ce que vous venez de faire, c'est attribuer les primaires de l'espace colorimétrique dans l'espace CIE 1931 XYZ, en fonction des points du triangle représentant le gamut Rec2020. Bien que les données soient linéaires, lorsque vous voyez l'image sur votre écran, la transformation de l'espace colorimétrique d'affichage lui est appliquée, en utilisant l'intention de rendu que vous avez choisie.

Cela dit, il n'est pas vraiment nécessaire d'attribuer un profil de couleur à ce stade. Vous utiliserez très probablement le mode de visualisation Auto-ajustement lorsque vous travaillerez avec des données linéaires, ce qui ne nécessite de toute façon pas de gestion des couleurs.

Maintenant que c'est expliqué, vous pouvez poursuivre tranquillement le reste de votre flux de travail linéaire. Calibrage, alignement, empilement, suppression d'étoiles, réduction de bruit... Faites-vous plaisir !

6.8.2 Étape non linéaire

Lorsque vous êtes prêt à étirer votre image, il est temps de repenser à votre espace colorimétrique. L'étirement transforme les données linéaires en données non linéaires afin que l'image soit agréable à l'œil humain. Vous allez maintenant rendre vos données non linéaires, donc avant l'étirement, c'est le bon moment pour convertir l'image dans l'espace colorimétrique non linéaire de votre choix, que ce soit sRGB ou Rec2020 ou un autre espace colorimétrique de votre préférence. Vous pouvez soit le faire vous-même manuellement, soit définir une préférence pour Siril, soit pour vous inviter à convertir l'espace colorimétrique de l'image en votre espace colorimétrique préféré, soit pour le faire automatiquement.

Vous pouvez maintenant poursuivre et terminer toute retouche post-étirement de votre image.

6.8.3 Sauvegarde et exportation d'images

Lorsque vient le temps de sauvegarder votre image, vous avez le choix entre plusieurs formats :

— Le format FITS est le format d'image natif de Siril et les profils ICC peuvent être intégrés dans les fichiers FITS. Cela se fait de la même manière que les autres logiciels d'astrophotographie, de sorte que les images enregistrées

avec des profils ICC intégrés seront interopérables entre Siril et PixInsight (et probablement d'autres logiciels prenant en charge l'intégration de profils ICC dans les fichiers FITS).
- TIFF et PNG sont de bons formats à choisir pour l'exportation d'images de haute qualité et offrent une plus grande compatibilité avec d'autres logiciels que le format FITS. Les deux prennent en charge l'intégration de profils ICC. Les options de la boîte de dialogue *Préférences* permettent de définir si les images TIFF et PNG d'une profondeur de 8 bits et plus, sont enregistrées au format sRGB, enregistrées dans le profil ICC actuel de l'image ou enregistrées dans l'espace colorimétrique préféré défini dans les préférences.
- La norme *de facto* pour les images hautement compressées pour le WWW est JPEG. La prise en charge des profils ICC dans les images JPEG varie en fonction de votre bibliothèque JPEG, mais si vous utilisez libjpeg-turbo version 2.0.0 ou supérieure, les profils ICC seront intégrés. Pourtant, de nombreux programmes tiers ne prennent pas du tout en charge la gestion des couleurs et supposeront que les fichiers JPEG représentent une image dans l'espace colorimétrique sRGB avec la norme sRGB TRC. Il s'agit de l'espace colorimétrique d'exportation par défaut pour les fichiers JPEG et vous devez y réfléchir attentivement avant de les enregistrer avec un profil de couleur différent.
- Siril prend également en charge d'autres formats d'image : PNM/PGM, BMP etc. Ces formats sont regroupés sous la catégorie « ne supportent pas les profils ICC ». Lors de l'exportation vers ces formats, les images seront **toujours** converties en sRGB.

Siril fournit un paramètre de préférence pour l'intention d'exportation/conversion. Dans presque tous les cas, vous devez laisser cela comme « Colorimétrie relative ».

6.9 Profils d'espace couleur

6.9.1 Profils de couleurs intégrés

Siril contient un ensemble de profils d'espace colorimétrique de base. Ceux-ci sont tirés de l'excellent dépôt d'Elle L. Stone de profils de couleurs libres et bien conçus. Son site web contient une mine d'informations techniques sur la théorie et la pratique des espaces colorimétriques, et une grande source d'inspiration pour l'application du traitement des couleurs aux images. Tous les profils qui s'y trouvent sont publiés sous la licence Creative Commons Attribution-Share-Alike Unported, version 3.0.

Les profils intégrés sont destinés à fournir un ensemble suffisant d'espaces colorimétriques pour la plupart des usages : sRGB pour l'exportation vers le web et comme profil d'affichage par défaut, Rec2020 comme profil à large gamut qui semble pouvoir devenir la prochaine norme pour les moniteurs à large gamut de haute qualité, et bien sûr une gamme de profils de Gris pour correspondre aux profils RVB.

- sRGB
  - Profile linéaire sRGB_elle_V4_g10
  - Profile CRT sRGB sRGB_elle_V4_srgbtrc
  - Profil ICC sRGB de préférence avec tables de référence perceptuelles (pour l'affichage uniquement)
  - Profile sRGB TRC sRGB_elle_V2_srgbtrc (pour exporter)
- Rec2020
  - Profile linéaire Rec2020_elle_V4_g10
  - Profile Rec709 TRC Rec2020_elle_V4_rec709
  - Profile Rec709 TRC Rec2020_elle_V2_rec709 (for export)
- Gris
  - Profile linéaire Gray_elle_V4_g10
  - Profile TRC sRGB Gray_elle_V4_srgbtrc
  - Profile sRGB TRC Gray_elle_V2_srgbtrc (pour exporter)
  - Profile Rec709 TRC Gray_elle_V4_rec709
  - Profile Rec709 TRC Gray_elle_V2_rec709 (pour exporter)

Notez que les versions CRT natives des profils sont fournies en deux versions (version 4 et version 2). La version 4 offre de meilleures fonctionnalités (en particulier pour les images à grande profondeur de bits), notamment des CRT paramétriques qui évitent la quantification. Cependant, les profils de la version 4 ne sont pas encore universellement supportés. Par conséquent, lorsque vous exportez un fichier pour l'utiliser dans un autre logiciel, il est plus sûr d'incorporer un profil V2.

Le profil Gris est disponible avec des CRT correspondant aux deux espaces colorimétriques RVB intégrés, ainsi qu'avec une CRT linéaire.

6.9.2 Profiles CRT linéaires

Les données du capteur sont au départ une représentation linéaire de la lumière, alors pourquoi n'attribuons-nous pas un profil de couleur linéaire à des images non étirées telles qu'une image nouvellement empilée ? Techniquement, nous devrions peut-être le faire. Il est certain que si vous souhaitez le faire, vous pouvez le faire, et rien de mal ne se produira. Vous pouvez le faire en utilisant la boîte de dialogue de gestion des couleurs, et il existe également une préférence (Attribution rigoureuse de profils ICC linéaires) qui attribue une version linéarisée du profil ICC de travail lorsqu'un profil est attribué automatiquement à une image nouvellement chargée (si elle ne présente aucun signe d'étirement préalable), à une image empilée ou à une image nouvellement composée. Cependant, il n'y a généralement aucun avantage à procéder de la sorte. Pour comprendre pourquoi, nous devons comprendre une différence fondamentale entre l'astrophotographie et le traitement photographique normal.

En photographie normale, l'ensemble de l'image est bien exposé, avec des zones d'ombre plus sombres et des zones de lumière plus claires. Au stade brut, l'image est convertie de linéaire à un profil de travail, mais cette opération est réversible et, en fait, fréquemment inversée, car de nombreuses opérations d'édition (mélange des couleurs, réduction du bruit, etc.) sont mieux réalisées en mode linéaire.

Cependant, en astrophotographie, tous les détails sont profondément ancrés dans les ombres et nous devons appliquer un étirement très fort pour produire une image agréable à regarder. Cet étirement est beaucoup plus extrême que le léger changement de gamma de 1.0 à 2.2 pour passer de la lumière linéaire à sRGB, et **il n'est pas facilement réversible** - en particulier si plusieurs étirements sont appliqués. Nous devons donc toujours effectuer le même type d'opérations (correction des couleurs, réduction du bruit, déconvolution, etc.) sur la lumière linéaire (et beaucoup d'autres choses qui n'existent même pas dans la photographie normale devraient également être effectuées sur les images linéaires : modélisation des étoiles, élimination des étoiles, etc.)

Il n'est pas nécessaire de définir un profil de couleur linéaire pour effectuer ces opérations - les algorithmes tels que la réduction du bruit, etc. ne sont même pas conscients des profils de couleur. Ils s'appliquent simplement aux données fournies.

Il est donc possible d'assigner le profil de couleur à une image à n'importe quel stade de l'édition, bien qu'il soit généralement plus judicieux de le faire juste avant l'étirement. Ce qu'il faut retenir, c'est qu'en astrophotographie, **toutes les opérations à effectuer sur les images linéaires doivent être faites avant l'étirement**. En général, l'étirement et les derniers réglages de la balance des couleurs, de la saturation, etc. devraient être la dernière opération d'édition d'une image. Et à ce stade, vous devez éditer l'image dans l'espace colorimétrique que vous avez choisi, ce qui donnera un aspect aussi cohérent que possible à l'image lorsqu'elle sera visualisée dans n'importe quelle application de gestion des couleurs ou sur n'importe quel périphérique de sortie.

6.9.3 Recommandations en matière d'espace colorimétrique

Cette section explique un peu plus en détail pourquoi Siril propose les espaces colorimétriques intégrés qu'il propose.

sRGB est la norme *de facto* pour le web. Comme mentionné ci-dessus, si vous voulez vous assurer que vos images sont bonnes dans tous les navigateurs web, ainsi que dans toutes les applications tierces qui peuvent ou non prendre en charge la gestion des couleurs, vous devez vraiment exporter en sRVB. C'est une bonne solution pour la plupart des moniteurs SDR, et en l'absence d'une gestion des couleurs appropriée, les images qui ont un aspect correct dans les versions précédentes de Siril doivent être considérées comme étant en sRVB.

Cependant, considérons à nouveau l'espace colorimétrique CIE 1931. La forme en fer à cheval qui forme la limite de l'espace colorimétrique représente les *teintes spectrales pures*. Si vous disposez d'une source lumineuse parfaitement monochromatique, comme un laser ou un de ces lampadaires jaunes au sodium, vous regardez la limite de la CIE 1931. En tant qu'astrophotographes, ces sources monochromatiques pures sont en fait très importantes pour nous, surtout si vous faites de l'imagerie à bande étroite. Ces couleurs ne peuvent pas être représentées avec précision dans le système sRGB. La proportion des couleurs visibles qui peuvent être représentées dans sRGB est en fait assez faible. Nous pouvons faire mieux.

Alors, si un gamut étendu est meilleur, pourquoi l'espace colorimétrique à large gamme intégré dans la norme Siril est Rec2020 plutôt qu'un espace colorimétrique encore plus étendu ? AllColorsRGB, ACES 2065 et ProPhoto RGB sont tous beaucoup plus grands - ACES 2065 peut représenter toutes les parties du spectre visible.

Le problème est qu'ils y parviennent en fixant leurs couleurs primaires - les valeurs "100 % rouge", "100 % vert" et "100 % bleu" - *en dehors* de la gamme du visible. Cela pose un problème, en particulier pour les imageurs à bande étroite qui souhaitent attribuer des filtres aux couleurs primaires : vous vous retrouvez avec des couleurs imaginaires dans votre image et vous devez toujours compter sur l'intention d'une transformation de l'espace colorimétrique pour faire ce qu'il faut en les transformant en quelque chose de visible. Certains de ces espaces colorimétriques sont également linéaires, comme l'ACES 2065. C'est une bonne chose pour les artistes CGI, mais pas pour nous, en raison de la lenteur des transformations d'affichage lorsque l'on travaille avec des espaces linéaires. Siril peut optimiser certaines transformations à partir d'espaces colorimétriques linéaires en plus de ce que fait lcms2, mais seulement si les primaires RVB sont les mêmes, par exemple si l'on transforme un sRGB linéaire en sRGB g22). La norme Rec2020 a donc été choisie car elle fournit un espace colorimétrique non linéaire avec le gamut le plus large sans avoir de primaires imaginaires.

Si vous n'êtes pas encore convaincu par un espace colorimétrique particulier, je vous recommande d'essayer Rec2020 comme espace colorimétrique pour l'édition, ainsi que pour l'envoi d'images à des services d'impression de haute qualité qui peuvent gérer les images en couleur, et sRGB pour l'exportation vers le web. Mais il existe aussi d'autres bons choix disponibles dans les fichiers de profil ICC que vous pouvez utiliser à la place, comme Adobe RGB et ProPhoto RGB (ROMM), si vous préférez.

6.9.4 Profils de couleurs tiers

Profils de moniteur Siril inclut le profil de moniteur sRGB v4 de l'ICC avec la prise en charge des tables de correspondances de rendu perceptuel et l'utilise par défaut. Un autre profil de moniteur peut être utilisé à la place, mais notez que seules les intentions supportées par un profil donné seront disponibles - de nombreux profils sRGB largement disponibles ne supportent que l'intention Colorimétrique Relative.

Profils d'épreuvage en douceur Il existe une grande variété de normes de presse et de papiers dans le monde et chaque combinaison nécessite son propre profil ICC. Siril ne peut pas fournir tous ces profils. Par conséquent, pour utiliser le mode d'affichage de l'épreuvage à l'écran, vous devez fournir le profil ICC d'épreuvage à l'écran approprié dans la fenêtre Préférences.

Il se peut que vous souhaitiez travailler dans un espace colorimétrique autre que ceux intégrés, par exemple ProPhoto RGB. Cela est possible, mais vous devrez fournir vous-même les profils ICC. La fenêtre des préférences propose des contrôles pour définir un profil RVB et un profil gris avec une CRT correspondant (essentiellement, avec le même gamma). Si vous avez l'intention d'exporter des fichiers dans l'espace colorimétrique que vous avez choisi, il est recommandé que le profil que vous fournissiez soit un profil ICC V2, pour une compatibilité maximale avec d'autres logiciels.

6.9.5 Limites de l'affichage HDR

Les espaces colorimétriques d'affichage à large gamut tels que Rec2100 avec HLG ou PQ TRC peuvent nécessiter des tampons de pixels de plus de 8 bits pour s'afficher de manière fluide. Malheureusement, Siril utilise la bibliothèque graphique Cairo pour l'affichage et Cairo ne peut pas encore traiter les tampons de pixels de plus de 8 bits. L'impact de ce problème est probablement négligeable pour la plupart des utilisateurs. Cependant, si vous avez la chance d'utiliser un écran HDR >1000nit capable d'afficher des espaces colorimétriques étendus et que vous utilisez certaines combinaisons d'images tierces et de profils de moniteur personnalisés, vous pourrez alors constater des artefacts mineurs de montée en escalier des couleurs pour certaines images. Il s'agit uniquement d'un problème d'affichage qui n'affectera pas l'aspect imprimé de votre image, ni même son aspect lorsqu'elle est transformée en un espace colorimétrique plus étroit ou visualisée sur des systèmes d'exploitation qui prennent en charge les tampons de pixels à grande profondeur de bits.

Pour l'instant, il n'y a rien à faire, mais si Cairo ajoute à l'avenir la prise en charge de la profondeur de bits élevée, il sera possible d'améliorer la situation.

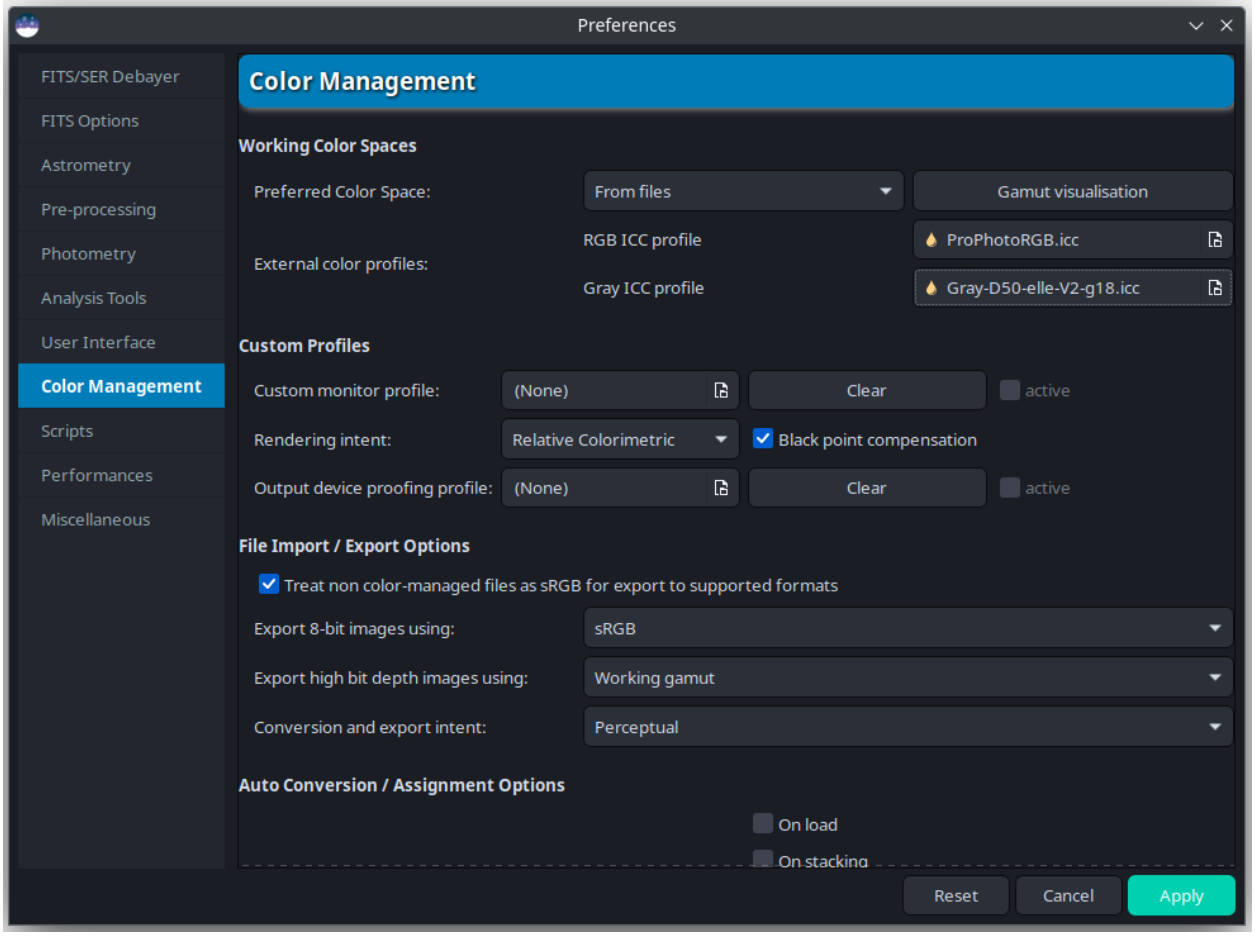


FIG. 4 – L'image montre comment configurer un espace de travail ProPhoto en utilisant les profils d'Elle Stone. Le gamma standard de ProPhoto RGB (ROMM) est de 1,8, donc en plus du profil ProPhotoRGB, nous ajoutons le profil d'Elle Stone Gris avec gamma = 1,8. Vous pouvez télécharger tous les profils d'Elle Stone ici. Si vous n'avez pas le profil ProPhotoRGB, vous pouvez également utiliser le profil "LargeRGB-elle-v2-g18" d'Elle Stone, qui est exactement le même, sauf qu'elle a évité d'utiliser le terme "ProPhoto RGB" pour d'éventuelles raisons de droits d'auteur.

6.10 L'épreuvage en douceur

6.10.1 Épreuve d'affichage en douceur

Siril prend en charge l'épreuvage numérique. Cela permet de visualiser, dans l'espace couleur de votre moniteur, l'aspect de votre image lorsqu'elle sera imprimée via un standard spécifiée. Pour utiliser l'épreuvage à l'écran, vous devez spécifier un profil d'épreuvage à l'écran dans la boîte de dialogue Paramètres.

Si aucun profil de périphérique de sortie n'est spécifié, ou s'il est rendu inactif en utilisant la case à cocher des préférences, l'épreuvage écran utilisera le profil d'affichage comme cible d'épreuvage. Ceci active la vérification de l'image *gamut* dans le menu Vérifications de l'image, qui montre les pixels qui sont en dehors du gamut de couleurs de votre écran. Lorsque la vérification du gamut est active, les pixels en dehors du gamut sont affichés en magenta vif.

6.10.2 Exportation pour impression

Siril ne permet pas d'exporter en format CMJN à des fins d'impression. La plupart des services d'impression de photos s'attendent à ce que les images soient fournies au format RVB et les pilotes d'imprimante font un bien meilleur travail de conversion RVB en un mélange des encres spécifiques utilisées par chaque imprimante qu'une simple conversion RVB vers CMJN au niveau de l'application. Si vous ne savez pas exactement pourquoi vous devez convertir une image en CMJN, il est presque certain que vous n'avez pas besoin de le faire. Cependant, afin de minimiser l'écrêtage du gamut lorsque votre service d'impression convertit l'image en un profil CMJN, il vaut la peine d'utiliser un profil RVB à large gamut tel que le Rec2020 intégré.

6.11 Changements dus à l'introduction de la gestion des couleurs

Siril gère désormais correctement les couleurs.

Cela signifie qu'il ne le faisait pas vraiment correctement dans le passé. Siril n'était pas une exception à cet égard : de nombreux programmes ne fournissent pas de gestion des couleurs, mais partent du principe que tout est fait dans un espace colorimétrique sRGB. Cela convient la plupart du temps, mais des problèmes ont été signalés lors de l'échange d'images entre Siril et des applications graphiques gérant correctement les couleurs.

Certains utilisateurs ont signalé que dans certaines circonstances, les images sauvegardées par Siril avaient un aspect différent dans d'autres logiciels. Pour les images exportées à partir du nouveau Siril avec gestion des couleurs, cela devrait cesser. Les images auront un aspect cohérent dans tous les logiciels avec une mise en œuvre correcte de la gestion des couleurs.

Cependant, si vous avez des images étirées enregistrées en FITS dans une version précédente de Siril et que vous les ouvrez dans Siril avec gestion des couleurs, elles peuvent dans certains cas paraître pâles et délavées. Pas de panique ! Voici une explication de la raison pour laquelle cela se produit, et comment cela peut très facilement être évité / corrigé.

6.11.1 Pourquoi cela se produit-il ?

Lorsque Siril lit une image FITS sans profil ICC intégré, si les préférences "Attribution automatique du profil ICC" sont activées, il essaie de déterminer si l'image a été étirée ou non. Il ne peut le faire qu'en vérifiant les entrées HISTORY dans l'en-tête FITS. S'il détecte que des fonctions d'étirement ont été appliquées, il suppose que l'image a été étirée sur un moniteur sRGB et applique le profil correspondant. Cependant, les anciennes versions de Siril n'ajoutaient pas toujours l'historique pour chaque commande, et d'autres logiciels peuvent ou non ajouter l'historique, mais Siril ne peut pas reconnaître tous les mots-clés d'étirement qui peuvent être utilisés par d'autres logiciels, de sorte qu'il peut être incapable de détecter qu'une image a été étirée. L'hypothèse de secours que nous choisissons de faire est qu'une image FITS sans profil ICC intégré est une image **linéaire**.

Cependant, votre FITS étiré n'est pas vraiment une image linéaire. Les données ont été étirées, et vous les avez en fait étirées pour qu'elles aient l'air correctes dans l'espace colorimétrique de votre moniteur, qui est probablement approximativement sRGB.

Lorsqu'un profil ICC linéaire lui est attribué, les routines d'affichage appliquent alors une transformation de l'espace colorimétrique de l'espace linéaire au profil d'affichage, ce qui a pour effet de l'étirer une seconde fois.

6.11.2 Comment l'éviter ?

Ce problème ne se produit que si l'option "Attribution automatique de l'espace colorimétrique préféré au chargement" est activée. Si cette option est désactivée, Siril n'attribuera aucun profil de couleur lors du chargement du fichier. Vous pouvez alors en assigner un manuellement à l'aide de la boîte de dialogue Gestion des couleurs.

6.11.3 Comment régler le problème ?

Si vous avez chargé une image avec l'option Assignation automatique activée et que le mauvais profil est assigné (ou si le mauvais profil est assigné pour toute autre raison, par exemple si vous avez chargé un fichier sauvegardé par un autre logiciel qui a assigné un mauvais profil), il est également très facile de résoudre le problème. Il suffit d'ouvrir l'image, d'aller dans la boîte de dialogue Gestion des couleurs et de sélectionner "sRGB (standard sRGB TRC)" dans la liste déroulante (ou "Gray (sRGB TRC)" si l'image est mono). Cliquez sur le bouton *Assigner* (**pas** *Convertir vers*) et Siril assignera le profil de couleur sélectionné à votre image.

Le tour est joué. L'affichage sera désormais correct et lorsque vous enregistrerez l'image, le profil de couleur approprié sera incorporé.

CHAPITRE 7

Séquences

Les séquences sont ce que Siril utilise pour représenter un ensemble de fichiers manipulés, par exemple l'ensemble des images darks que nous allons transformer en master dark. C'est un outil très utile pour manipuler un grand nombre de fichiers qui doivent être liés les uns aux autres.

7.1 Un ensemble de deux fichiers FITS ou plus

Siril utilise nativement des données en virgule flottante 32 bits ou des données entières non signées 16 bits pour les images *FITS*, les autres formats sont automatiquement convertis. Pour être reconnus et détectés comme une séquence, les noms de fichiers des images FITS doivent respecter un modèle particulier qui est :

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

— **basename** peut être n'importe quoi en utilisant des caractères ascii. Il est généralement pratique, mais pas obligatoire, qu'il se termine par le caractère _. Il sera utilisé comme nom de séquence.
— **$i** est l'index de l'image. Cela doit être un nombre positif et peut commencer par plusieurs zéros.
— **[ext]** est l'extension supportée comme expliqué dans les *références*, `fit` par défaut.

Note

L'extension utilisée pour détecter les séquences FITS dans le répertoire de travail actuel sera la même que celle configurée dans les paramètres et que celle des fichiers créés par Siril.

Avertissement

Certains systèmes d'exploitation limitent le nombre d'images qui peuvent être ouvertes en même temps, ce qui est nécessaire pour les méthodes d'empilement médian ou moyen. Pour Windows, la limite est de 2048 images. Si vous avez beaucoup d'images, vous devez utiliser un autre type de séquence, décrit ci-dessous.

7.2 Un seul fichier SER

Le *SER* est un format destiné à contenir une séquence d'acquisition de plusieurs images contiguës dans un seul fichier. C'est un format assez simple qui ne peut pas contenir autant de métadonnées que FITS, mais plus que de simples films et les données ne sont pas compressées. Les fichiers SER ne peuvent contenir que des images de 8 ou 16 bits par canal. Il existe trois types de fichiers SER, en fonction du contenu en pixels : monochrome, CFA ou couleur (3 canaux).

Note

Un fichier SER peut être ouvert soit via *Fichier* et *Ouvrir*, soit avec le bouton *Chercher séquences*.

Voir *ici* pour plus d'informations sur le format SER et pourquoi les formats de film comme AVI non compressé ne devraient pas être utilisés pour l'astronomie.

Avertissement

Dans une certaine mesure, un fichier de film ordinaire tel que AVI ou tout autre conteneur est également pris en charge. Le support des fichiers film est abandonné au profit de SER, mais il peut toujours être utile d'ouvrir un film dans Siril, pour explorer son contenu, extraire quelques images ou les convertir. Quelques opérations peuvent encore être effectuées, mais de manière plus lente qu'avec d'autres séquences, comme l'empilement par somme. Pour un traitement complet, vous serez confronté à des limitations et des incompatibilités.

7.3 Un seul fichier FITS

Note

Également appelés cubes FITS ou séquences FITS, ou FITSEQ en abrégé dans Siril.

Le format FITS est un conteneur d'images et de données scientifiques, il peut en contenir plusieurs dans un seul fichier. Nous pouvons l'utiliser pour stocker une séquence entière d'images FITS dans un seul fichier tout en préservant l'en-tête FITS de chaque image. Il s'agit du format de fichier utilisé par les astronomes professionnels.

Il est plus simple de gérer un seul fichier sur le disque que 2000, mais comme il s'agit d'un seul fichier, certaines opérations sur les images uniques de la séquence peuvent ne pas être possibles. En particulier, Siril ne permet pas actuellement de modifier l'en-tête d'une seule image.

Ce format est une alternative au SER pour une séquence à fichier unique, avec 32 bits par canal et un support complet de l'en-tête.

7.4 Charger un séquence

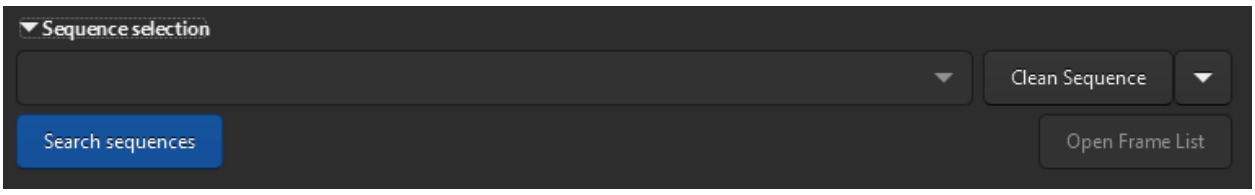


Fig. 1 – Recherche et nettoyage de séquences.

Lorsque le répertoire de travail est placé au bon endroit, que les FITS suivent la bonne nomenclature, et que l'extension des fichiers FITS est également correctement définie, cliquez alors sur le bouton *Chercher séquences* de l'onglet Sequence. Une liste déroulante s'ouvre avec toutes les séquences disponibles dans le dossier. Si une seule est trouvée, elle est automatiquement sélectionnée et chargée.

7.5 Sélecteur d'image

Une grande force de Siril est qu'il permet de manipuler facilement des séquences d'images. Lorsqu'une séquence est ouverte, l'image de référence (voir ci-dessous) s'affiche, par défaut il s'agit de la première image. Cependant, il peut parfois être utile d'inspecter les images individuelles d'une séquence. Ceci est possible avec le sélecteur d'images, disponible via la barre d'outils avec le bouton ou via l'onglet de la séquence avec le bouton *Ouvrir la liste des images*.

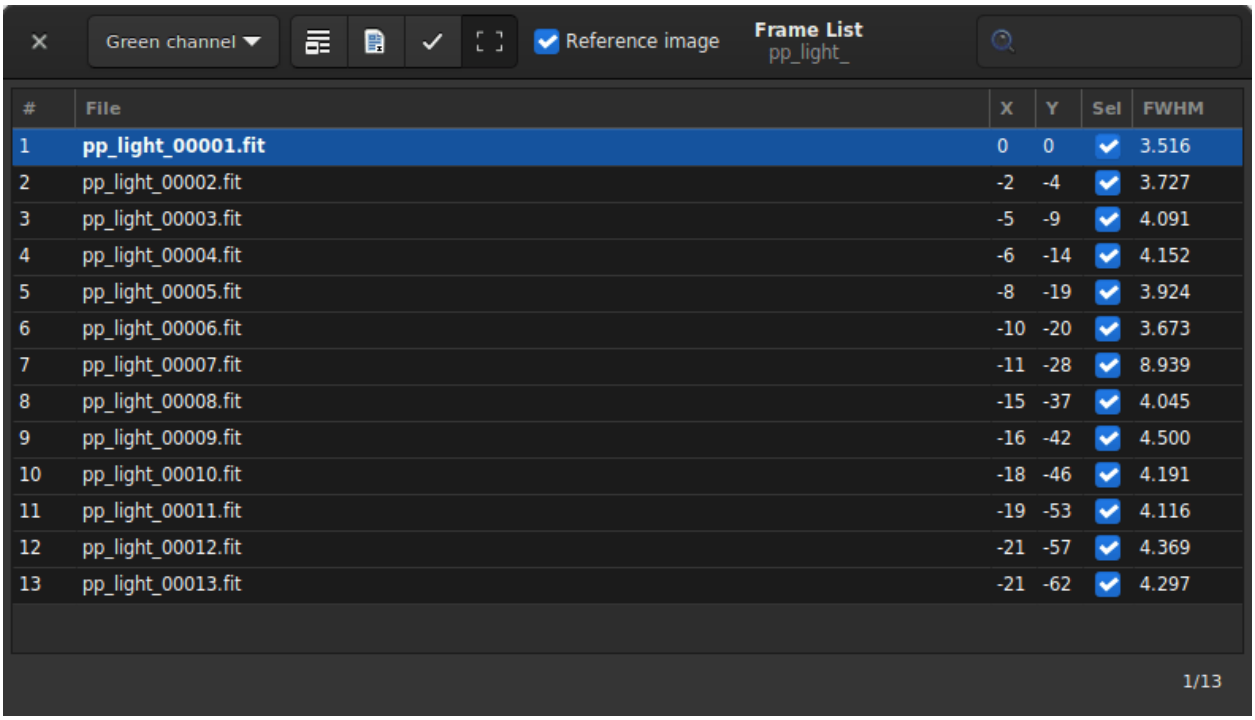


Fig. 2 – Sélecteur d'images qui vous permet de choisir une image dans la séquence et de l'afficher, de la définir comme référence ou de l'exclure.

Astuce

En survolant une image dans le sélecteur d'image, le nom du fichier d'origine s'affiche, s'il a été enregistré.

En cliquant sur une image dans la liste, celle-ci sera chargée et affichée dans la zone principale, tout en conservant la séquence comme objet actif pour le traitement. Plus qu'un simple sélecteur d'affichage d'images, l'outil peut également être utilisé pour **exclure manuellement des images** de la séquence, ou visualiser celles qui sont encore incluses, visualiser les valeurs de qualité d'image et de décalage entre les images si elles ont été calculées, et changer l'image de référence. Notez que plus d'informations sur la qualité de l'image peuvent être visualisées dans l'onglet Plot.

L'exclusion d'une image de la séquence ne signifie pas que ses données seront définitivement supprimées, mais simplement qu'elle ne sera pas utilisée pour les opérations de traitement ultérieures, si on le lui demande. Dans la plupart

des cas, l'option à rechercher s'appelle *Traiter uniquement les images incluses*.

L'**image de référence** est l'image de la séquence qui servira de cible pour l'alignement et la normalisation. Les autres images seront transformées pour ressembler à l'image de référence, elle doit donc être choisie avec soin. Heureusement, depuis Siril 1.2, un nouvel algorithme d'alignement en deux passes permet de sélectionner automatiquement la meilleure image de la séquence comme image de référence avant de procéder à la transformation de l'image.

La barre d'en-tête de la fenêtre fournit de nombreux contrôles pour ces propriétés de séquence :

— Le menu déroulant permet de modifier le canal pour lequel les données d'alignement (qualité, décalages) sont affichées, si elles existent pour d'autres canaux.
— Le premier bouton de la barre d'outils définit toutes les images de la séquence comme étant exclues manuellement.
— Le second, les définit tous comme inclus.
— La troisième inclut ou exclut les images sélectionnées dans la liste (les sélections multiples peuvent être faites avec `Ctrl` ou `Shift`) de la séquence.
— Le dernier bouton peut être désactivé pour masquer l'indicateur de cadrage sur les images aligné.

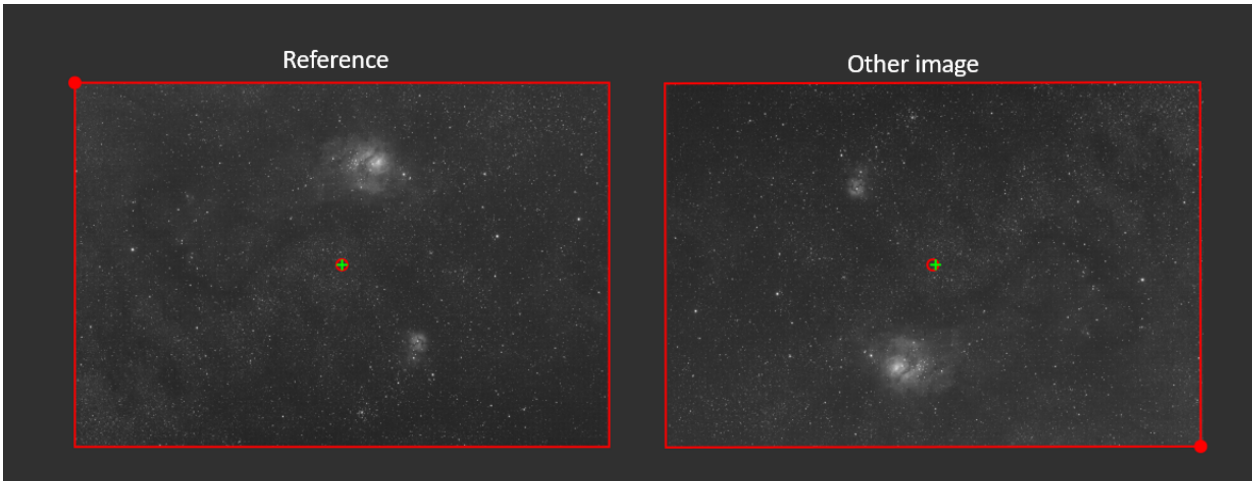


FIG. 3 – Indicateur d'image

The red frame shows the trace of the reference frame over each image of the sequence. The filled circle in one corner indicates the position of the top-left corner of the reference image. For instance, for the right image above, the framing indicator circle is at the bottom right, showing that there has been a meridian flip between this image and the reference shown on the left.
The red circle/green cross close to the center show the position of the center of the reference image and of the current image respectively. This enables to visualize the shift between the two images.

Note

Since Siril 1.3.1, the color of the framing indicator is different whether the registration information is just pure shift (in blue) or more *complex transformations* (in red for linear transformations, in green when distortions are included). For previous versions, it was always red, irrespective of the transformation.

— Le bouton *Image de référence* est utilisé pour sélectionner l'image de référence pour la séquence. Toutes les séquences doivent en avoir une, ce sera la première image si elle n'est pas définie ou par défaut.
— Enfin, le champ de recherche vous permet de trouver les images par leur nom.

Il est également possible de trier toutes les images en cliquant sur les en-têtes de colonne. Vous pouvez ainsi trier les images en fonction de leur nombre ou de leur FWHM. Cette dernière option est très utile pour voir les meilleures et les pires images.

7.5.1 Commandes de manipulation des séquences

Ligne de commande Siril

```
setref sequencename image_number
```

Définit l'image de référence de la séquence donnée en premier argument. **image_number** est le numéro séquentiel de l'image dans la séquence, et non le numéro dans le nom de fichier, commençant à 1

Ligne de commande Siril

```
select sequencename from to
```

Cette commande permet de sélectionner facilement en masse des images dans la séquence **sequencename** (de **from** à **to** inclus). Il s'agit d'une sélection en vue d'un traitement ultérieur.
Voir aussi UNSELECT

Liens : *unselect*

Ligne de commande Siril

```
unselect sequencename from to
```

Permet une sélection en masse des images dans la séquence **sequencename** (de **from** à **to** inclue). Voir SELECT

Liens : *select*

7.6 Exportation de la séquence

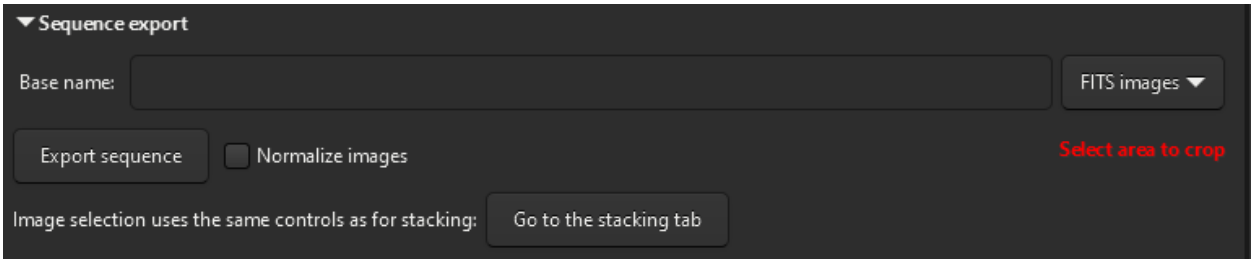


FIG. 4 – Donnez un nom et spécifiez le format de sortie pour exporter une séquence.

L'outil **Exportation de la séquence** vous permet d'exporter une séquence d'images dans différents formats. Il est particulièrement utile si vous souhaitez exporter les images en tenant compte des informations d'alignement contenues dans le fichier `seq`, avec un recadrage et une normalisation optionnels.

Avec la fonction d'exportation de séquence, vous pouvez sélectionner une séquence à exporter, choisir le format de

fichier et le niveau de compression pour les formats vidéo. La fonction d'exportation de séquences de Siril prend en charge un large éventail de formats de fichiers d'images, notamment *FITS* (fichier FITS unique ou fichier FITS de séquence), *TIFF*, *SER* , AVI, MP4 et WEBM et peut s'avérer pratique lors de la construction de timelapse.

Le bouton *Normaliser les images* permet de normaliser les images par rapport à l'image de référence. La normalisation est la même que celle effectuée *pendant l'empilement*, avec les paramètres suivants : *Additif avec mise à l'échelle*, *Normalisation plus rapide* désactivée.

De plus, il est possible de jouer avec les critères de *filtrage des images* pour exclure ou non les images en fonction de leur qualité. Un bouton *Aller à l'onglet Empilement* a été ajouté ici, pour aller facilement à l'onglet qui les expose.

7.7 Informations sur la séquence

Toutes les informations sur les séquences, transformation d'alignement, les statistiques et la sélection des images sont stockées dans un fichier `.seq` enregistré à côté des fichiers de séquences. Il est fortement recommandé de ne jamais éditer ce fichier manuellement car Siril y écrit continuellement et un seul caractère erroné pourrait corrompre la lecture de la séquence.

Une façon de nettoyer le contenu de ce fichier de séquence est d'aller dans l'onglet **Séquence** et de cliquer sur *Nettoyer Séquence*. Le choix de ce qui sera nettoyé peut être défini en cliquant sur la petite flèche à côté.

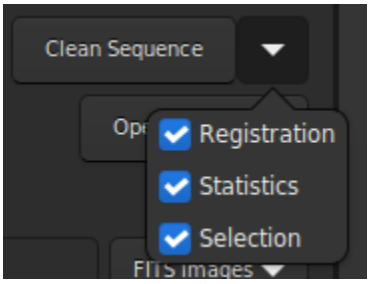


FIG. 5 – Menu pour nettoyer le fichier de séquence.

Ligne de commande Siril

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

Cette commande efface les données de sélection, d'alignement et/ou de statistiques stockées pour la séquence **sequencename**.

Vous pouvez préciser de n'effacer que l'alignement, les statistiques et/ou la sélection avec les options **-reg**, **-stat** et **-sel**, respectivement. Tout est effacé si aucune option n'est passée

CHAPITRE 8

Définitions et déroulement des opérations

L'astrophotographie est le processus de capture des images d'objets célestes. Elle comporte plusieurs étapes, dont le prétraitement et le traitement, qui sont distinctes mais liées.

Le *Prétraitement* est l'étape initiale du travail avec des données astrophotographiques brutes. Il s'agit de préparer ces données en vue d'un traitement ultérieur. Cette étape comprend généralement la soustraction du dark, la correction du flat et la correction d'autres problèmes de base tels que l'élimination des pixels chauds et froids.

Le *Traitement* désigne le post-traitement des données prétraitées, généralement après l'empilement. C'est là que l'astrophotographe applique diverses techniques pour améliorer l'image finale et faire ressortir les détails et les caractéristiques. Il peut s'agir d'un renforcement de la netteté (déconvolution), d'un étalonnage des couleurs, d'une réduction du bruit et d'un étirement de l'image afin d'accroître la visibilité des détails peu visibles.

En résumé, le prétraitement prépare le terrain pour le traitement en s'assurant que les données sont sous une forme appropriée et débarrassées des signaux indésirables, tandis que le traitement consiste à faire ressortir le meilleur du signal pour créer l'image finale. Les deux étapes sont importantes dans le processus d'astrophotographie, et la qualité du résultat dépend des compétences et des techniques appliquées à ces deux stades.

Dans Siril, le prétraitement principal est effectué en suivant l'ordre des onglets du panneau de droite et nécessite l'utilisation de fichiers maîtres. C'est un processus qui peut être automatisé assez facilement et les scripts fournis dans Siril effectuent cette tâche. Le traitement des images se fait via le menu dédié *Traitement de l'image*. Ce processus est plus difficile à automatiser car il est spécifique à chaque image et consiste en un travail itératif.

CHAPITRE 9

Prétraitement

Cette section vous présente les différentes étapes du prétraitement de vos images, de l'importation dans Siril à l'obtention d'une image empilée.

Le volet de droite contient les onglets qui sont utiles pendant le prétraitement. Ils ont été conçus pour être utilisés de gauche à droite tout au long du processus, avec quelques exceptions pour la création de masters. Ces onglets sont également accessibles via les touches `F1` à `F7`.

Le prétraitement est l'étape avant l'alignement et l'empilement des images. L'objectif est de supprimer tous les signaux indésirables et de réduire le bruit présent sur toutes les sous images.

9.1 Conversion

Siril supporte le format FITS 32bits ainsi que le format SER de manière native. Par conséquent, tout autre format de fichier doit d'abord être converti dans ces formats pour être pris en charge et générer une *séquence*. Le type de fichiers pris en charge est indiqué dans l'onglet et dépend de la manière dont Siril a été compilé.

Siril présente un onglet de conversion qui est divisé en 2 panneaux. Le panneau supérieur vous permet de charger les fichiers **source** que vous souhaitez convertir.

La gestion de ces fichiers se fait à partir de la mini barre d'outils .

— Le premier bouton, le bouton +, est celui qui permet de charger les fichier sources. Il ouvre une fenêtre de dialogue vous permettant de choisir tous les fichiers à convertir présent sur votre ordinateur. Seuls les formats supportés par Siril sont visibles.

> **Astuce**
>
> Il est possible de glisser et déposer des fichiers directement dans la zone **sources**. La zone de dépôt est mise en évidence lorsque les fichiers sont au-dessus d'elle.

— Le second bouton, le bouton -, permet de supprimer les fichiers sélectionnés. Plusieurs fichiers peuvent être supprimés en même temps. Ils ne sont pas supprimés du disque dur, mais seulement de la zone de conversion.
— Le dernier bouton vous permet de supprimer tous les fichiers chargés en une seule fois.

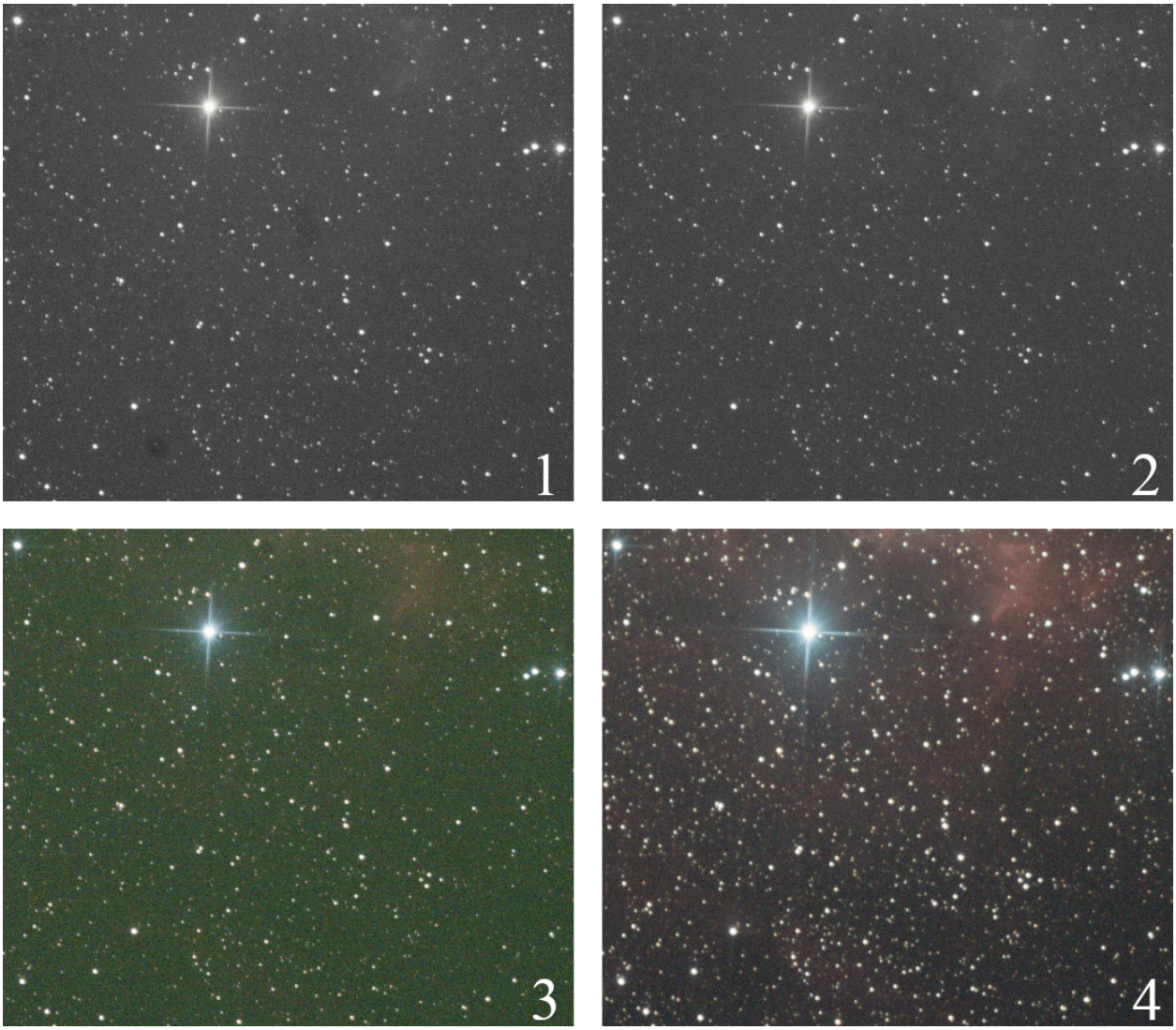


FIG. 1 – **L'image 1** montre le résultat de la conversion d'une image brute d'un appareil photo numérique. Vous pouvez voir des poussières visibles, semblables à des taches sombres. **L'image 2**, après l'étalonnage des images par les master darks, bias et flats, montre la suppression complète de ces taches et un signal plus propre. **L'image 3** est la même après dématriçage, montrant la couleur et une très grande dominante verte due à la plus grande sensibilité des photosites verts sur les capteurs. Enfin, **l'image 4** est la sortie de l'empilement, avec équilibrage des canaux.

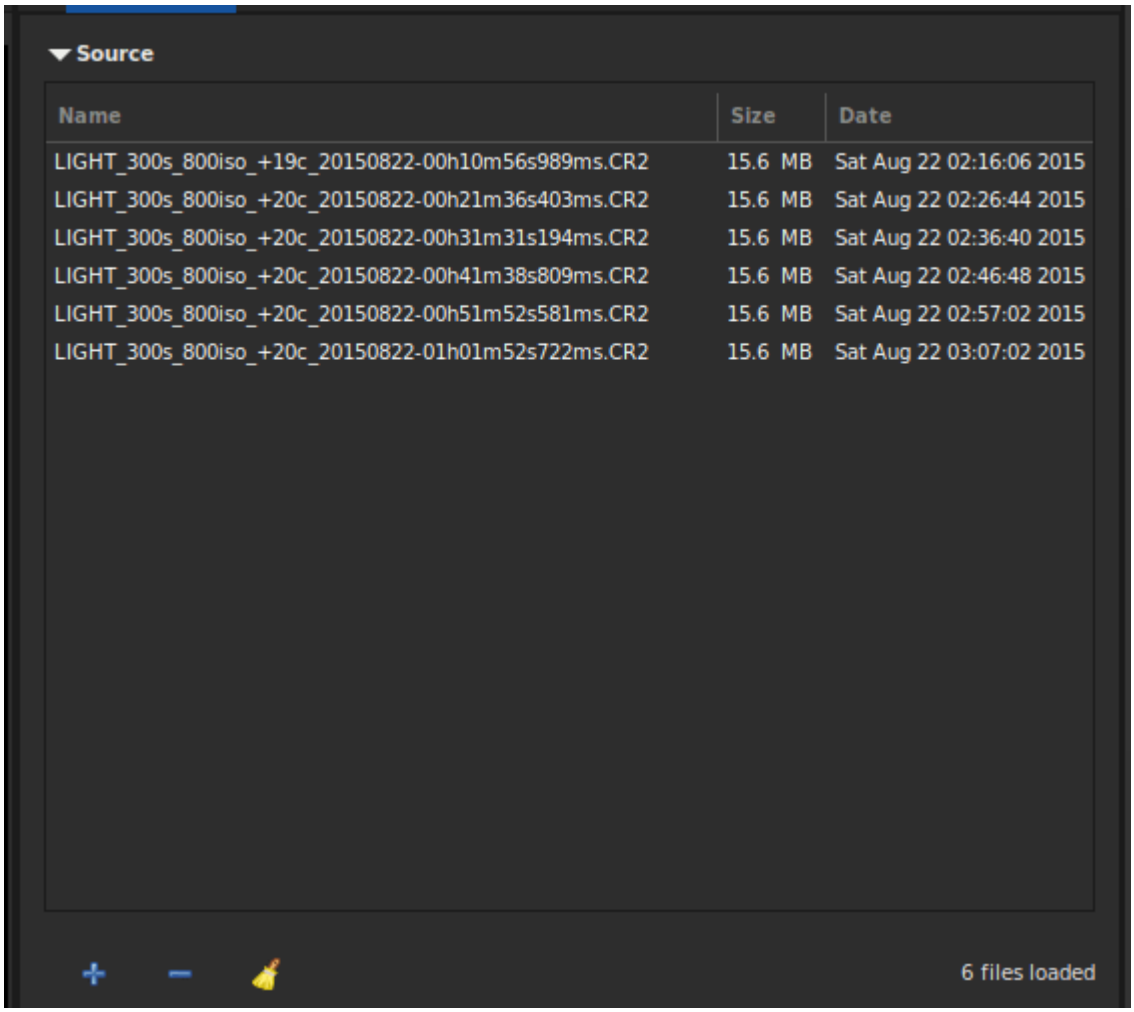


FIG. 2 – Panneau Source de l'onglet conversion.

Le nombre de fichiers chargés et sélectionnés est indiqué dans la barre d'état, à droite de la barre d'outils.

Dans la section **destination** il est possible de choisir le nom de la séquence qui sera générée après la conversion des fichiers.

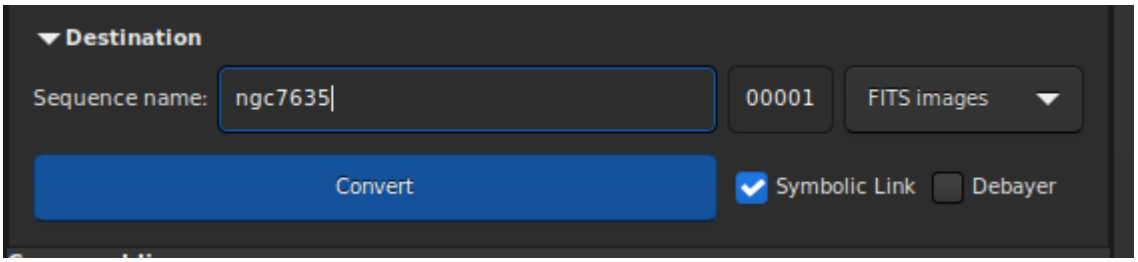


FIG. 3 – Panneau Destination de l'onglet conversion.

Ainsi pour un nom de séquence `basename`, les fichiers convertis seront de la forme

```
basenameXXXXX.[ext]
```

L'extension est tel que définie dans les *préférences*. L'indice `XXXXX` commence par défaut à `00001` avec la première image, mais il est possible de définir un indice de départ différent. Cela peut être utile dans le cas d'une multisession qui partage les mêmes fichiers maîtres. Trois types de sorties sont possibles, à choisir dans le menu déroulant :

- images FITS
- séquence SER
- séquence Fits

Ces formats de fichiers sont expliqués dans la section *séquence* de cette documentation.

Techniquement, lorsque les fichiers d'entrée sont au format FITS, il n'est pas nécessaire de les convertir. Cependant, on peut vouloir le faire pour que les fichiers soient renommés pour créer une séquence et puissent être traités dans Siril. Afin de ne pas remplir inutilement le disque dur, il est alors possible de choisir l'option *Lien symbolique*. Cette option crée un lien symbolique pour les fichiers FITS au lieu de les copier. Cette option n'est donc disponible que lorsque les fichiers de sortie sont des images FITS.

Note

Lorsque les liens symboliques sont activés, cela désactive la compression.

Avertissement

Pour Microsoft Windows, pour utiliser les liens symbolique il faut activer le mode développeur de Windows.

Avertissement

Si sous GNU/Linux vous voyez l'erreur **Symbolic link Error : Fonction non implémentée**, c'est peut-être parce que vous essayez de créer une séquence de liens symboliques dans un répertoire d'un système de fichiers qui n'autorise pas les liens symboliques.

Lorsque les formats de sortie sont des séquences SER ou FITS, l'option *Séquences multiples* devient visible. Cochez cette option pour créer plusieurs fichiers de séquences au lieu d'un seul fichier SER ou FITS pour tous les éléments

d'entrée. Utilisez cette option si les éléments d'entrée (fichiers de séquence tels que les films, les SER ou FITS cubes) ne partagent pas la même taille d'image ou ne doivent pas être traités ensemble.

La dernière option *Dématricer* permet à l'utilisateur de dématricer les images pendant la conversion. Cette option ne devrait généralement pas être utilisée si les images sont des offsets, des darks, des flats, ou des images destinées à être prétraitées. En effet, en raison de la prise en compte de la matrice de Bayer, le résultat RVB de votre image RAW est une image interpolée. Par conséquent, le prétraitement de données interpolées donnera des résultats erronés. La conversion des fichiers RAW d'un capteur couleur donne des images FITS monochromes Color Filter Array (Matrice de filtres colorés) (CFA). Contrairement aux images RVB, les images CFA représentent l'ensemble des données du capteur avec le motif de Bayer. L'image suivante montre un recadrage d'une image CFA. Notez que le motif de Bayer (RGGB sur cet exemple) est visible.

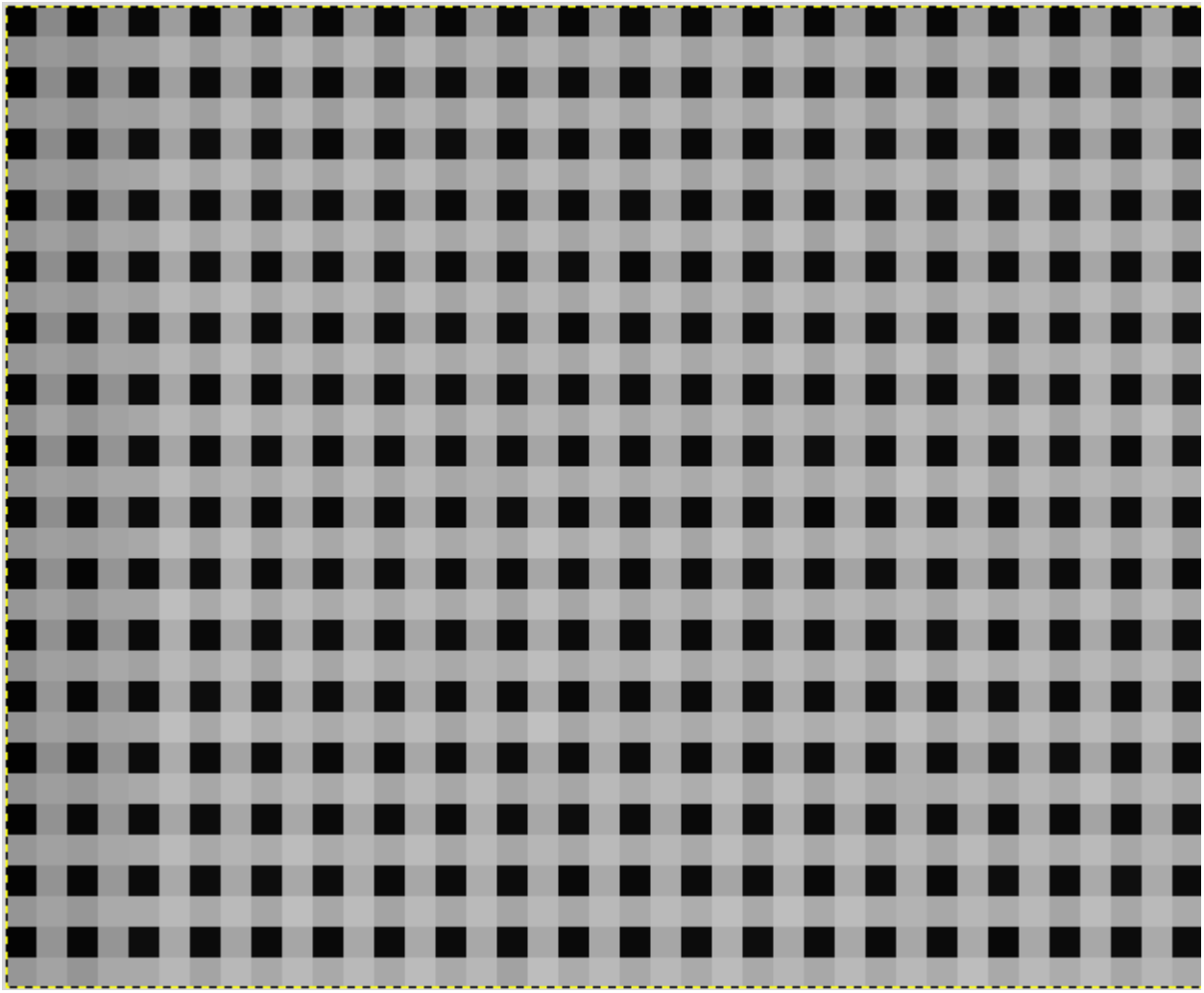


FIG. 4 – Modèle de Bayer montré sur une image CFA (Color Filter Array, Matrice de filtres colorés).

Enfin, le bouton *Convertir*, permet, comme son nom l'indique, de lancer la conversion des fichiers.

Note

Les images brutes des reflex numériques dépendent du fabricant et sont généralement des formats fermés. Par conséquent, le décodage de ces fichiers est une tâche complexe qui doit être effectuée par un code dédié. Pour Siril,

la tâche de conversion des fichiers bruts est effectuée par LibRaw. En fait, si un format de fichier, généralement récent, n'est pas lu, il faut regarder sur le site de LibRaw s'il est supporté. Si ce n'est pas le cas, le fait de leur fournir un fichier brut peut aider l'équipe de développement à le faire. Cependant, il est également possible que la version de LibRaw intégrée dans le paquet Siril ne soit pas la plus récente. Dans ce cas, vous devez soit attendre une nouvelle version, soit compiler les sources directement.

9.1.1 Fichier de correspondance

Après chaque conversion, un fichier se terminant par `_conversion.txt` est créé. Il contient les correspondances entre images d'entrée et images de la sequence obtenue lors de la conversion.

Ligne de commande Siril

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

Convertit toutes les images du répertoire de travail actuel qui sont dans un format supporté en séquence d'images FITS de Siril (plusieurs fichiers) ou en séquence FITS (un seul fichier) si **-fitseq** est fourni ou en séquence SER (un seul fichier) si **-ser** est fourni. L'argument **basename** est le nom de base de la nouvelle séquence, les nombres et l'extension seront mis derrière.

Pour les images FITS, Siril essaiera de faire un lien symbolique ; si ce n'est pas possible, les fichiers seront copiés. L'option **-debayer** applique le dématriçage aux images d'entrée CFA ; dans ce cas, aucun lien symbolique n'est effectué.

-start=index définit le numéro d'index de départ, utile pour continuer une séquence existante (non utilisé avec -fitseq ou **-ser** ; assurez-vous de supprimer ou d'effacer le .seq cible s'il existe dans ce cas).

L'option **-out=** modifie le répertoire de sortie en fonction de l'argument fourni.

Voir aussi CONVERTRAW et LINK

Liens : *convertraw*, *link*

Ligne de commande Siril

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

Identique à CONVERT mais ne convertit que les fichiers RAW des reflex numériques se trouvant dans le répertoire de travail actuel

Liens :*convert*

9.2 Calibration

Une fois la séquence chargée, les images peuvent être calibrées, alignées et empilées. La calibration est une étape facultative, mais importante, qui implique des images de bias, des images darks et des images flats. L'étalonnage d'une séquence dans Siril ne peut se faire qu'avec un bias maître, un dark maître et un flat maître, qui doivent d'abord être créés à partir de leurs séquences.

9.2.1 Fichiers maîtres

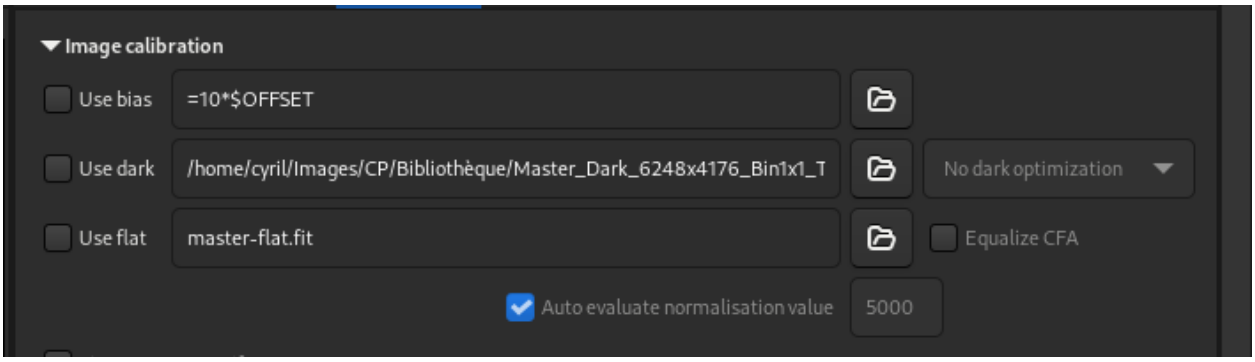


Fig. 5 – Paramètres des fichiers maîtres de l'onglet Calibration

Bias

Citons A Glossary of CCD terminology pour expliquer ce qu'est une image bias :

Le niveau d'offset d'une image CCD est un décalage électronique induit artificiellement qui garantit que le convertisseur analogique-numérique (ADC) reçoit toujours un signal positif. Toutes les données CCD présentent un tel décalage qui doit être supprimé si l'on veut que les valeurs des données soient réellement représentatives des comptes enregistrés par pixel.

Pour utiliser les bias maîtres dans Siril, cliquez sur le bouton à droite de l'entrée de texte et parcourez vos fichiers pour sélectionner le bon maître. Vous pouvez même utiliser le master-bias d'une bibliothèque telle que définie dans les *préférences*.

Astuce

Les bias doivent être pris avec l'obturateur fermé et le temps d'exposition le plus court possible. En principe, cela correspond à une exposition de 1/4000 s sur les appareils photos numériques modernes.

Bias synthétique

Comme le signal d'offset est très uniforme sur les capteurs modernes, nous recommandons de le traiter comme une image à niveau constant. Cela présente l'avantage d'économiser de l'espace disque et de minimiser le bruit dans l'image finale. À cette fin, Siril dispose d'une fonctionnalité qui rend la chose très facile.

Lors du prétraitement de vos flats, au lieu de spécifier un masterbias, vous pouvez directement taper des expressions dans le sélecteur de dossier telles que :

```
=2048
```

ou si l'en-tête FITS contient le mot clé `OFFSET`,

```
=64*$OFFSET
```

Les signes = et $ sont obligatoires. Le niveau doit être donné en ADU (et non en flottant, même si vous travaillez en 32 bits).

Traduit dans le langage de script, cela s'écrit :

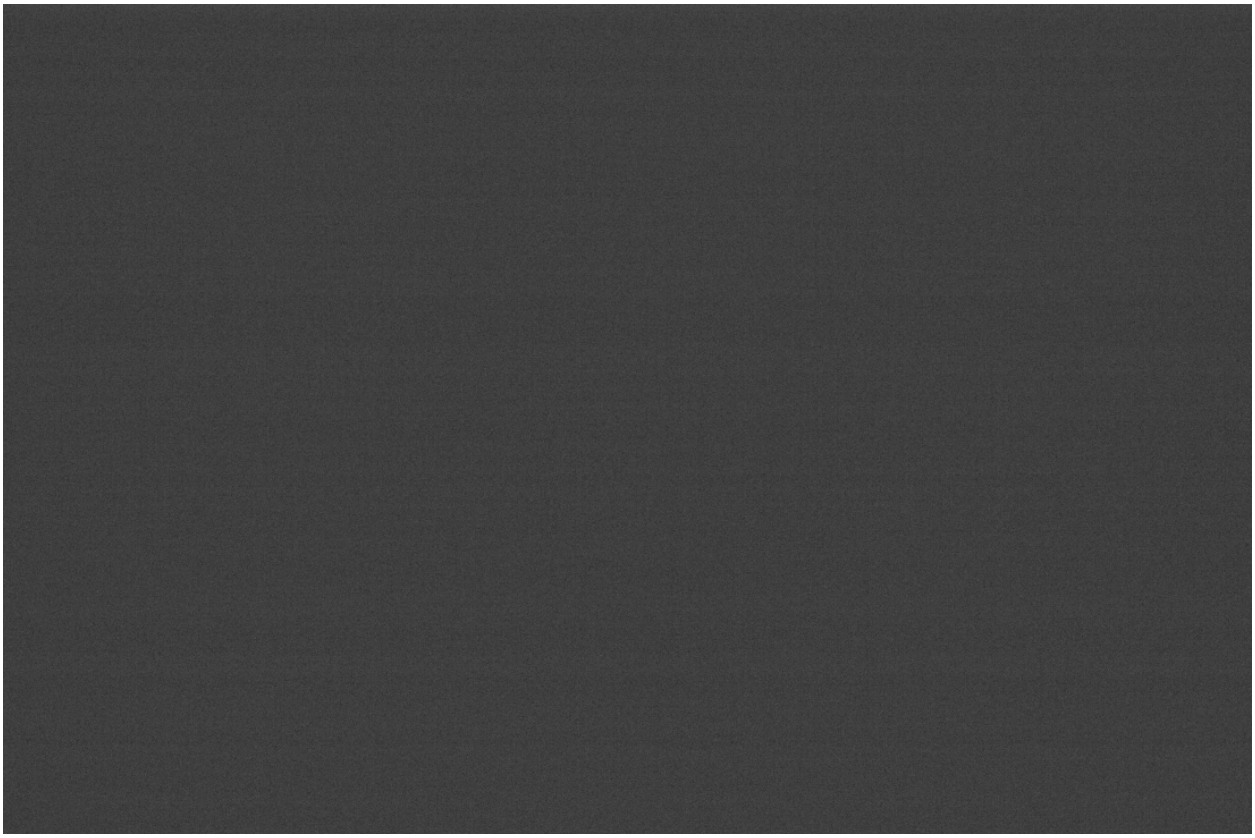


FIG. 6 – Exemple d'un bias pris avec un Canon EOS 1100D. Ne vous fiez pas au signal du bias légèrement visible, l'image est auto-étirée et les différences d'amplitudes des signaux sont très exagérées.

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

La valeur 2048 est ici un exemple pris pour les caméras dont le master-bias a une valeur médiane de 2048 .En général, pour les appareils photos numériques, la valeur est proportionnelle à une puissance de 2. Dans notre exemple, $2048 = 2^{11}$.

Pour plus de détails, veuillez vous référer au tutoriel sur les offsets synthétiques.

Dark

Les images darks contiennent le bruit thermique associé au capteur, le bruit étant proportionnel à la température et au temps d'exposition. Par conséquent, elles doivent être réalisées à peu près à la même température que les images lights. C'est la raison pour laquelle nous réalisons les images darks à la fin ou au milieu de la session d'imagerie.

Pour utiliser les dark maîtres dans Siril, cliquez sur le bouton à droite de l'entrée de texte et parcourez vos fichiers pour sélectionner le bon maître. Vous pouvez même utiliser le master-dark d'une bibliothèque telle que définie dans les *préférences*.

Astuce

Les darks sont réalisés au même temps d'exposition et ISO/Gain que les lights mais avec l'obturateur fermé.

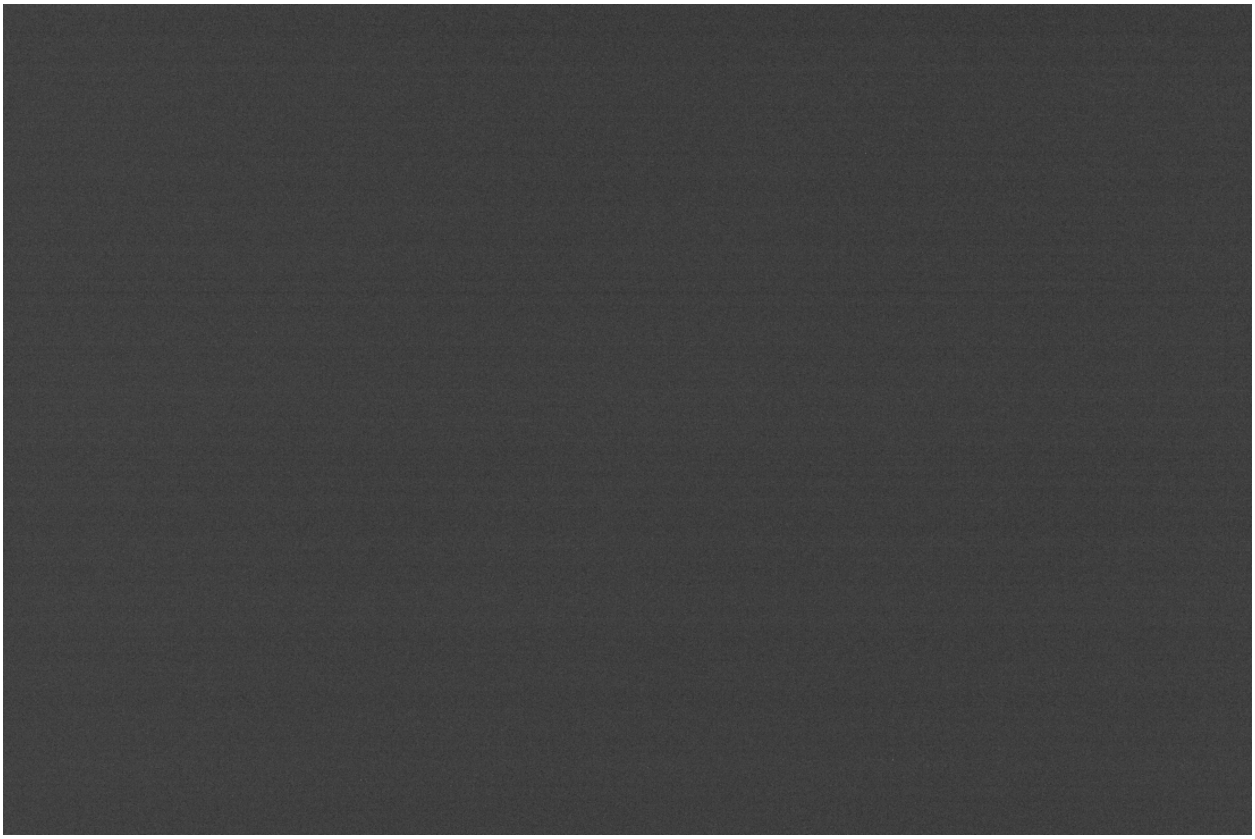


FIG. 7 – Exemple d'une image dark prise avec un Canon EOS 1100D avec 300s d'exposition et à ISO 800.

FIG. 8 – Une animation montrant la suppression du signal thermique grâce à la soustraction du dark.

Optimisation des dark

L'option d'optimisation de l'obscurité est utile lorsque les images darks n'ont pas été prises dans des conditions optimales. Siril propose deux méthodes avec des approches différentes, accessibles via une liste déroulante.

1. Avec l'option*Auto-évaluation*, la soustraction du dark peut être optimisée de façon à ce que le bruit de l'image résultante (image claire moins image sombre) soit minimisé en appliquant un coefficient à l'image dark.
2. La seconde option,*Utiliser exposition*, est basée sur les temps d'exposition des images si elles ont été enregistrées dans les mots-clés FITS.

Voici un exemple de situation où l'utilisation de cette option est nécessaire. Les images ont été prises avec une caméra **FLI ProLine 4240**. Le master-dark utilisé provient d'une bibliothèque et a été réalisé avec une exposition de 600s. Les images individuelles, quant à elles, ont des expositions de 60s. Le master-dark présente une signature de signal très particulière et assez disgracieuse : la présence de 4 préamplificateurs dans la caméra est à l'origine d'un tel signal. Ce défaut est évidemment aussi présent dans l'image de la galaxie, et la calibration par les darks doit être méticuleusement effectuée pour obtenir une image exempte de tout défaut.

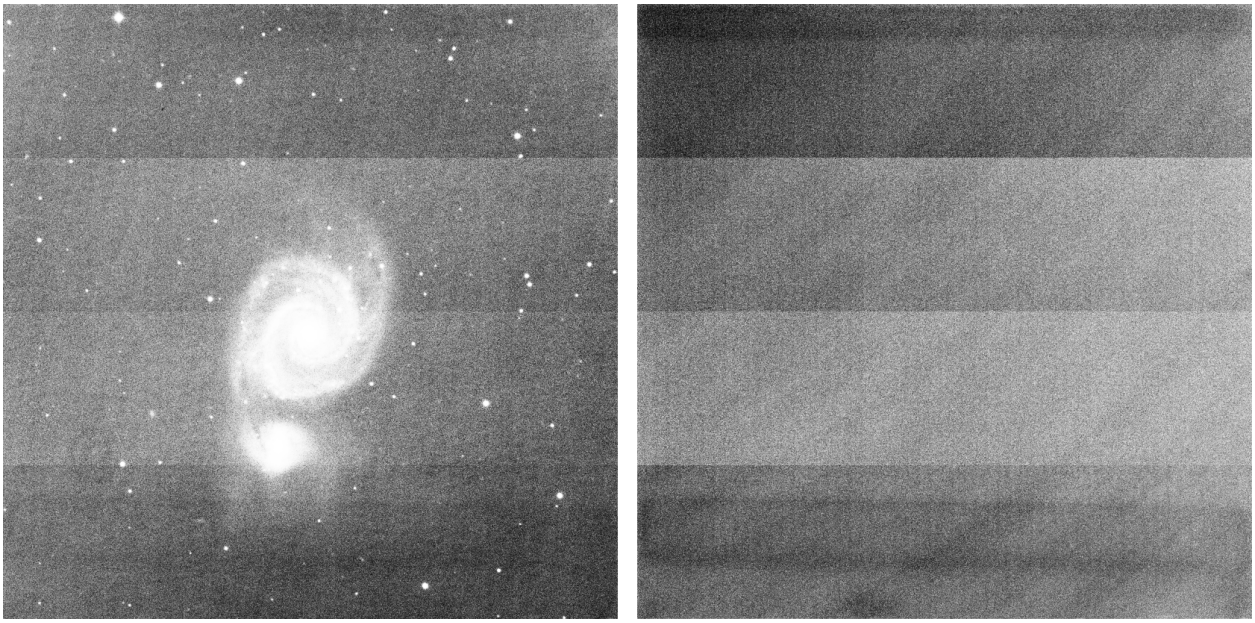


Fig. 9 – Il s'agit d'une image et du master-dark de la caméra **FLI ProLine 4240**. Vous pouvez voir 4 bandes très distinctes causées par les préamplificateurs, visible sur les deux types de fichiers. Les images sont affichées en mode égalisation d'histogramme, pour mettre en évidence les défauts.

Cependant, dans ce cas, si nous utilisons le flux de travail habituel, le résultat de la calibration sera très médiocre. En effet, le master dark n'a pas été pris dans les mêmes conditions d'exposition.

La solution est donc de soustraire le bias au dark, puis d'intégrer la soustraction du bias à celles des images, et de cocher la case d'optimisation des darks. Siril calculera automatiquement un coefficient à appliquer au fichier master-dark. Ici, il calcule **0.110**, ce qui est très cohérent, puisqu'il correspond à la différence d'un facteur 10 entre les darks et les images $(60/600 = 0.1)$.

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

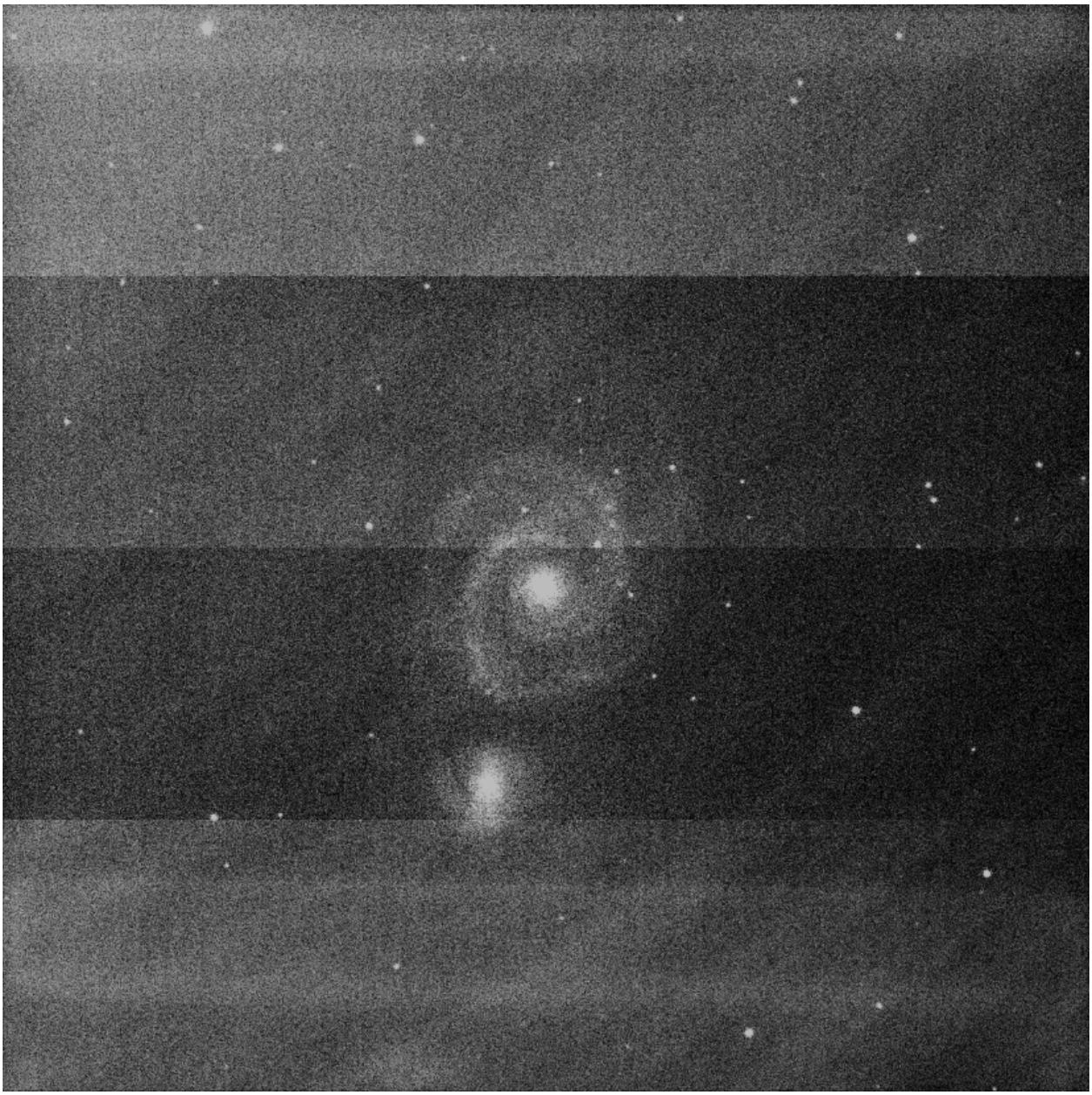


FIG. 10 – Avec le flux de travail classique, la calibration est médiocre et les défauts ne sont pas corrigés. L'image est affichée en mode d'égalisation d'histogramme, afin de mettre en évidence les défauts.

FIG. 11 – L'onglet calibration doit être complété comme il se doit dans un tel cas. Les master-flat et master-dark ont été calibrés par les bias.

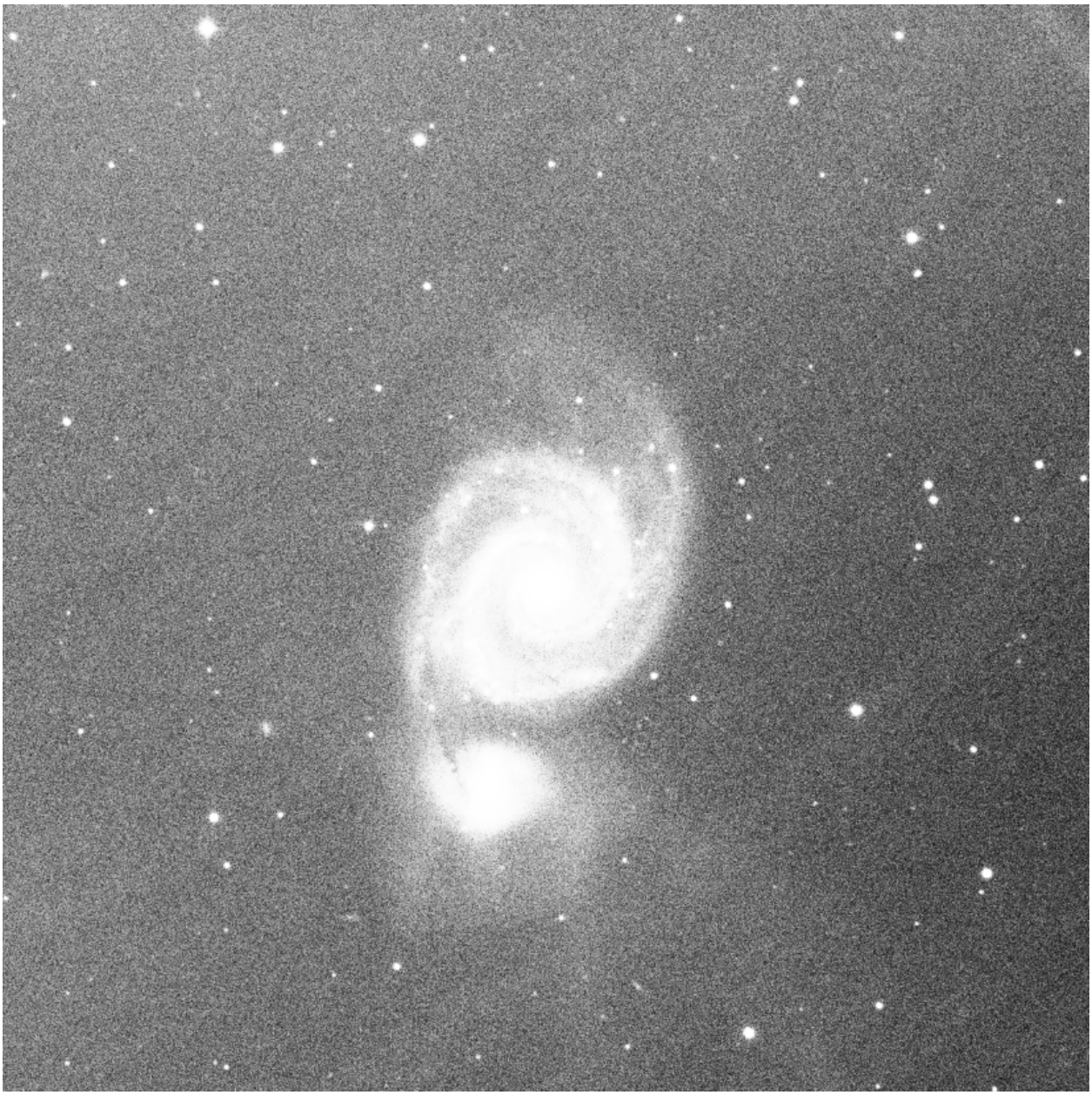


FIG. 12 – Grâce à l'optimisation des darks, la calibration est correcte. Le seul résidu visible est la frange du CCD dans le proche IR, qui ne peut être supprimée par la calibration. L'image est affichée en mode d'égalisation d'histogramme, afin de mettre en évidence les éventuels défauts.

Flat

Les télescopes n'éclairent généralement pas le détecteur de manière uniforme. En outre, la poussière sur les surfaces optiques et le capteur provoque des motifs plus sombres dans l'image résultante, et le capteur lui-même réagit différemment au nombre de photons qui frappent les différents photosites. Pour corriger ces effets, chaque image lumineuse doit être divisée par le master flat, qui devrait être la médiane des expositions individuelles d'une zone homogène et non saturée.

Pour utiliser les flates maîtres dans Siril, cliquez sur le bouton à droite de l'entrée de texte et parcourez vos fichiers pour sélectionner le bon maître. Vous pouvez même utiliser le master-flat d'une bibliothèque telle que définie dans les *préférences*.

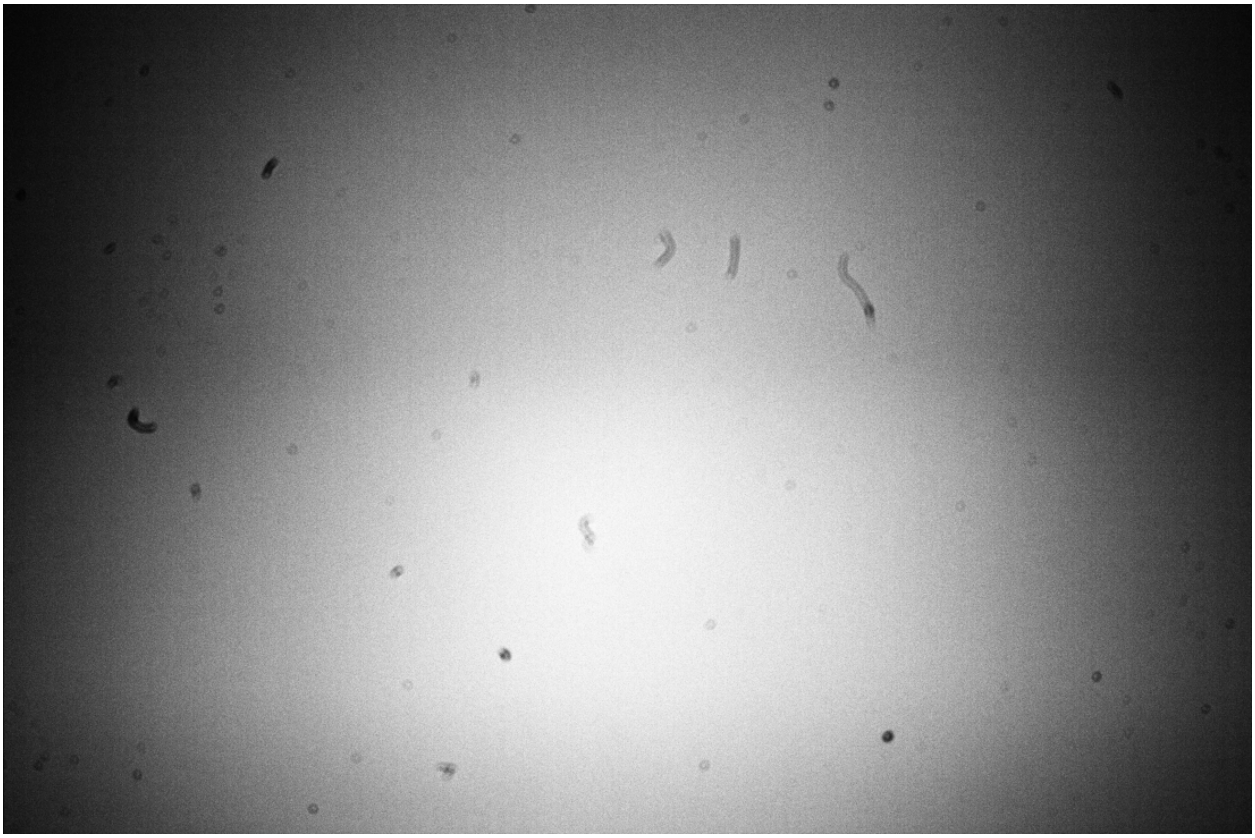


Fig. 13 – Exemple d'une image flat prise avec un Canon EOS 1100D. Les poussières présentes sur le chemin optique, et surtout sur le capteur, sont clairement visibles. Le vignettage (assombrissement des coins de l'image) est également très visible. Les défauts sont exagérés par le mode de visualisation. De plus, la commande *grey_flat* a été utilisée sur cette image pour faire disparaître le motif de Bayer.

Égalisation CFA

L'option *Égaliser CFA* égalise l'intensité moyenne des couches RGB d'une image flat CFA. Ceci est équivalent à l'utilisation de la commande *grey_flat*.

Ligne de commande Siril

```
grey_flat
```

Égalise l'intensité moyenne des couches RVB dans l'image CFA chargée. Il s'agit du même processus que celui utilisé sur les images flats lors de la calibration, lorsque l'option "equalize CFA" est utilisée

Auto-évaluation de la valeur de normalisation

Si l'option *Auto-évaluation de la valeur de normalisation* est cochée, Siril évaluera automatiquement la valeur de normalisation. Cette valeur est la moyenne du master-flat étalonné avec le master-bias. Sinon, c'est la valeur indiquée dans la zone de texte qui sera prise en compte.

9.2.2 Calibration des images lights

La calibration des images light consiste à appliquer le maître biais , dark et flat aux images astronomiques afin d'éliminer le signal indésirable.

Avertissement

En aucun cas l'étalonnage ne réduit le bruit des images. Au contraire, il l'augmente. C'est pourquoi il est important de prendre autant d'images d'étalonnage que possible, telles que des darks, afin de minimiser la quantité de bruit dans les images.

Corrige les artefacts des fichiers X-Trans

Cette option *Corrige les artefacts des fichiers X-Trans* permet de corriger les pixels de l'Auto Focus Fujifilm X-Trans. En effet, en raison du système de mise au point automatique à détection de phase, les photosites utilisés pour la mise au point automatique reçoivent un peu moins de lumière que les photosites environnants. L'appareil photo compense ce phénomène et augmente les valeurs de ces photosites spécifiques, ce qui donne un carré visible au milieu des images darks/offsets. Cette option n'a aucun effet sur les images de type Bayer. L'option n'est activée que si un master-bias ou un master-dark est chargé et utilisé.

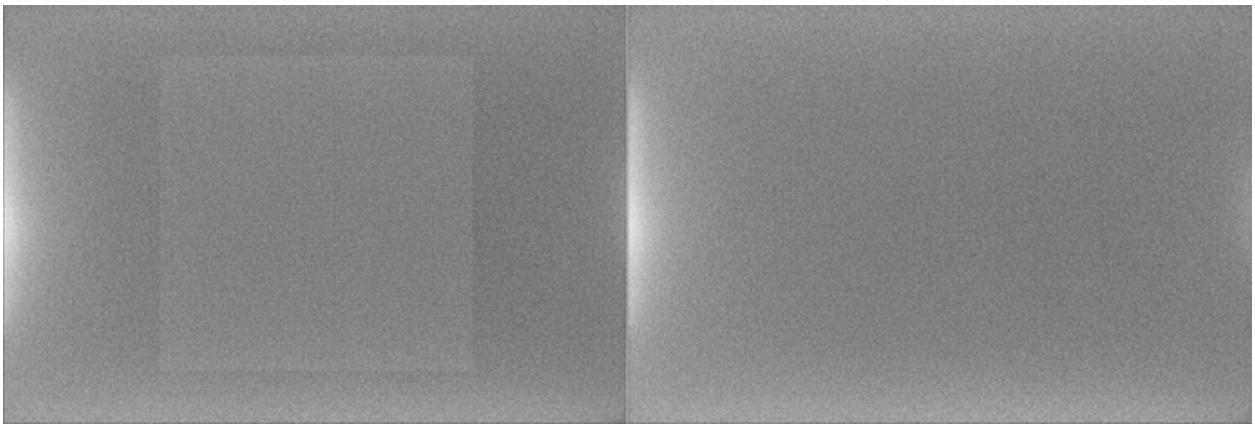


Fig. 14 – Artéfact X-Trans corrigé par l'algorithme de Siril

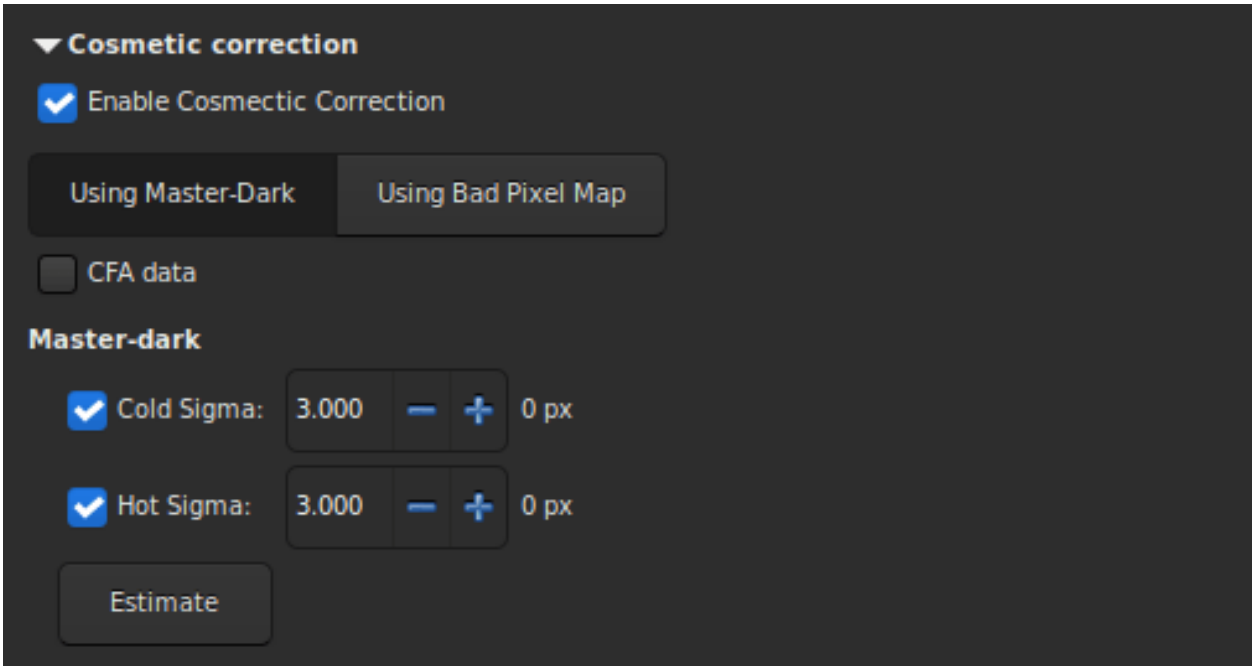
Correction cosmétique

La correction cosmétique est la technique utilisée pour corriger les pixels défectueux dans les images. En effet, tout capteur de caméra possède des photosites qui ne réagissent pas correctement à la réception des photons. Cela se traduit dans l'image par des pixels dont les valeurs sont très différentes de celles de leurs voisins les plus proches. Ces pixels sont appelés **pixels chauds**, si la valeur est beaucoup plus élevée, ou **pixels froids** lorsqu'elle est beaucoup plus faible.

Siril propose deux algorithmes pour corriger ces pixels défectueux si l'option *Activer la correction cosmétique* est cochée.

Utiliser le Master-Dark

Cette méthode requière la présence d'un master-dark. Siril recherchera les pixels dont l'écart par rapport à la médiane dépasse x fois l'écart-type σ. Cette valeur est ajustable pour les pixels chauds et froids.



Il est possible d'estimer le nombre de pixels qui seront corrigés dans l'image calibrée en appuyant sur le bouton *Estimer*. Si la valeur du pixel corrigé est affichée en rouge, cela signifie que ce nombre dépasse 1% du nombre total de pixels de l'image. Dans ce cas, vous devez augmenter la valeur du coefficient ou décocher la correction correspondante. Si les images proviennent d'un capteur de couleur, il est nécessaire de cocher l'option *CFA*.

Utiliser la Carte des Mauvais Pixels

Cette autre méthode utilise un fichier qui contient les coordonnées des pixels défectueux. Ce fichier est un simple fichier texte qui peut initialement être créé avec la commande*find_hot*. La dernière ligne a été ajoutée à la main et corrige une colonne endommagée située à la position$x = 1527$.

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

Ligne de commande Siril

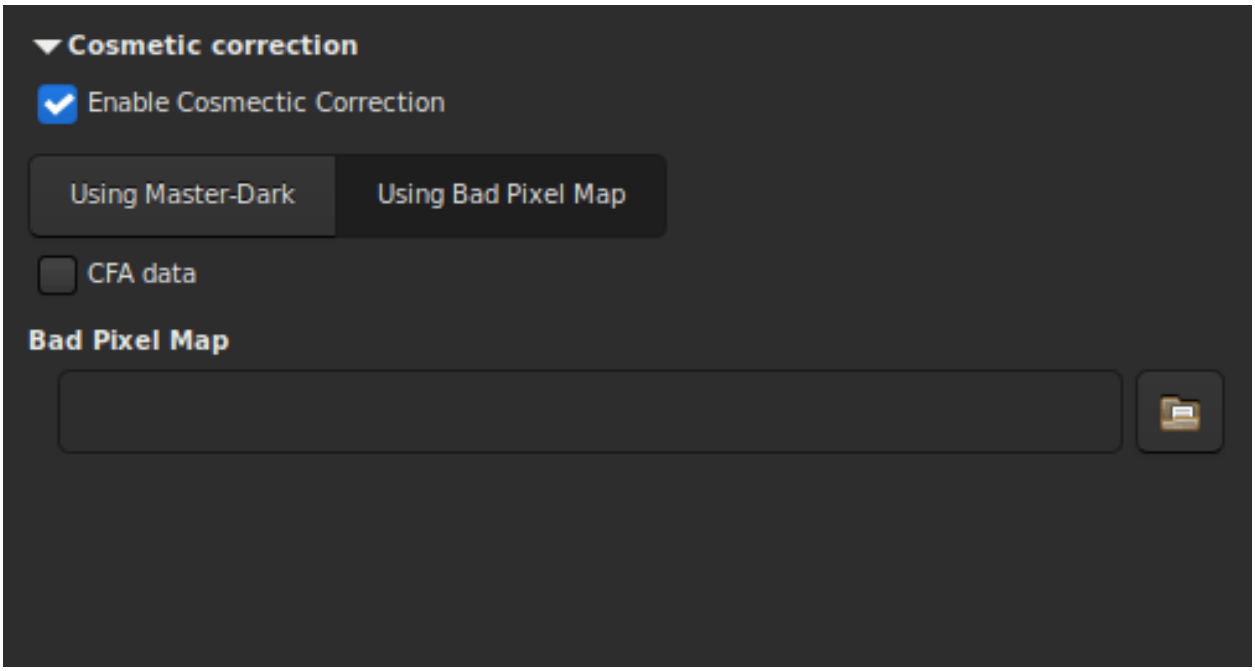
```
find_hot filename cold_sigma hot_sigma
```

Enregistre dans le répertoire de travail un fichier liste **filename** (format texte) contenant les coordonnées des pixels dont l'intensité **hot_sigma** fois supérieure et **cold_sigma** fois inférieure à l'écart-type, extraits de l'image chargée. Nous utilisons généralement cette commande sur un fichier dark maître. La commande COSME permet d'appliquer cette liste de mauvais pixels à une image chargée, voir aussi SEQCOSME pour l'appliquer à une séquence

Liens : *cosme*, *seqcosme*

Les lignes `P x y type` vont fixer le pixel aux coordonnées (x, y), où "type" est un caractère optionnel (C ou H) spécifiant à Siril si le pixel actuel est froid ou chaud. Cette ligne est créée par la commande FIND_HOT, mais vous pouvez également ajouter manuellement quelques lignes :
Les lignes `C x 0 type` vont fixer la colonne défectueuse aux coordonnées x.
Les lignes `L y 0 type` vont fixer la ligne défectueuse aux coordonnées y.

Ce fichier, qui peut être modifié à la main, doit être chargé comme une carte de mauvais pixels.

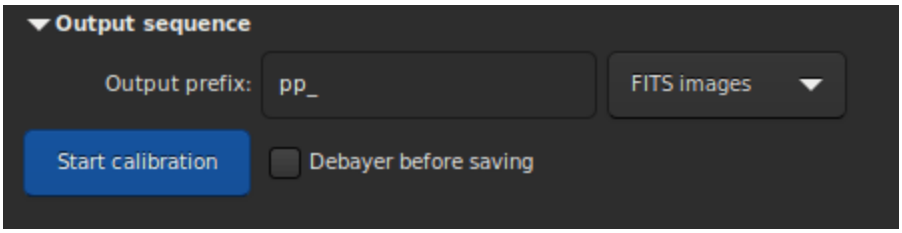


Enfin, si les images proviennent d'un capteur de couleur, il est nécessaire de cocher l'option *CFA*.

Séquence de sortie

Cette section regroupe les options qui peuvent être appliquées à la sortie.

- La zone de saisie *Préfixe de sortie* ajoute un préfixe aux images de sortie, afin de les identifier facilement. Par défaut, le préfixe est `pp_`, ce qui signifie pre-processed (pré-traitée).
- La liste déroulante définit le type de séquence de destination.
  - Images FITS : une image FITS par image.
  - Séquence SER : un fichier SER pour toute la séquence (limité à 16 bits par canal).
  - Séquence FITS : un fichier FITS pour toute la séquence.



— Dernière option, *Dématricer avant sauvegarde*. Cochez cette option si vous voulez appliquer un algorithme de dématriçage à vos images juste après qu'elles aient été calibrées. En faisant cela, vous sautez une étape manuelle qui peut prendre du temps.

Ligne de commande

Ligne de commande Siril

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

Calibre la séquence **sequencename** en utilisant les bias, dark et flat donnés en argument.

Pour les bias, un niveau uniforme peux être spécifié à la place d'une image, en entrant une expression entre guillemet commençant par le signe = tel que -bias="=256" ou -bias="=64*$OFSFSET".

Par défaut, la correction cosmétique n'est pas activée. Si vous voulez en appliquer une, vous devez le spécifier avec l'option **-cc=**.

Vous pouvez utiliser l'option **-cc=dark** pour détecter les pixels chauds et froids à partir du master dark (un master dark doit être fourni avec l'option **-dark=**), éventuellement suivi de **siglo** et **sighi** pour les pixels froids et chauds respectivement. Une valeur de 0 désactive la correction. Si les sigmas ne sont pas fournis, seule la détection des pixels chauds avec un sigma de 3 sera appliquée.

Alternativement, vous pouvez utiliser l'option **-cc=bpm** suivie du chemin vers votre carte de mauvais pixels pour spécifier quels pixels doivent être corrigés. Un fichier d'exemple peut être obtenu avec la commande *find_hot* sur un master dark.

Trois options s'appliquent aux images couleur (au format CFA) : **-cfa** pour les corrections cosmétiques, **-debayer** pour dématricer les images avant de les enregistrer, et **-equalize_cfa** pour égaliser l'intensité moyenne des couches RVB du flat maître, afin d'éviter de teinter l'image calibrée.

L'option **-fix_xtrans** est dédiée aux fichiers X-Trans en appliquant une correction sur les Darks et les Biais/offset pour supprimer un vilain motif carrée.

Il est également possible d'optimiser la soustraction des darks avec **-opt**, qui nécessite la fourniture de masters de bias et de dark, et calcule automatiquement le coefficient à appliquer au dark, ou calcule le coefficient grâce au mot-clé "exposure" avec **-opt=exp**.

Par défaut, les images marquées comme exclues ne sont pas traitées. L'argument **-all** peut être utilisé pour forcer le traitement de toutes les images même si elles sont marquées comme exclues.

Le nom de la séquence de sortie comme par le préfixe "pp_" sauf indication contraire spécifié avec l'option **-prefix=**.

Si **-fitseq** est fourni, la séquence de sortie sera une séquence FITS (fichier unique)

Ligne de commande Siril

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

Calibre l'image **imagename** en utilisant les bias, dark et flat donnés en argument.

Pour les bias, un niveau uniforme peux être spécifié à la place d'une image, en entrant une expression entre guillemet commençant par le signe = tel que -bias="=256" ou -bias="=64*$OFSFSET".

Par défaut, la correction cosmétique n'est pas activée. Si vous voulez en appliquer une, vous devez le spécifier avec l'option **-cc=**.
Vous pouvez utiliser l'option **-cc=dark** pour détecter les pixels chauds et froids à partir du master dark (un master dark doit être fourni avec l'option **-dark=**), éventuellement suivi de **siglo** et **sighi** pour les pixels froids et chauds respectivement. Une valeur de 0 désactive la correction. Si les sigmas ne sont pas fournis, seule la détection des pixels chauds avec un sigma de 3 sera appliquée.
Alternativement, vous pouvez utiliser l'option **-cc=bpm** suivie du chemin vers votre carte de mauvais pixels pour spécifier quels pixels doivent être corrigés. Un fichier d'exemple peut être obtenu avec la commande *find_hot* sur un master dark.

Trois options s'appliquent aux images couleur (au format CFA) : **-cfa** pour les corrections cosmétiques, **-debayer** pour dématricer les images avant de les enregistrer, et **-equalize_cfa** pour égaliser l'intensité moyenne des couches RVB du flat maître, afin d'éviter de teinter l'image calibrée.
L'option **-fix_xtrans** est dédiée aux fichiers X-Trans en appliquant une correction sur les Darks et les Biais/offset pour supprimer un vilain motif carrée.
Il est également possible d'optimiser la soustraction des darks avec **-opt**, qui nécessite la fourniture de masters de bias et de dark, et calcule automatiquement le coefficient à appliquer au dark, ou calcule le coefficient grâce au mot-clé "exposure" avec **-opt=exp**
Le nom du fichier de sortie comme par le préfixe "pp_" sauf indication contraire spécifié avec l'option **-prefix=**

9.2.3 Comprendre comment les flats corrigent les lights

Le but de cette section est de donner un peu plus de détails sur la façon dont les différents niveaux jouent un rôle dans la correction des lights par les flats.

Nous ne tiendrons pas compte ici des considérations relatives au bruit (encore une fois, le bruit ne disparaît pas avec la soustraction ou la division des maîtres, il diminue en calculant la moyenne de plusieurs réalisations du même processus aléatoire). Nous ne tiendrons pas compte non plus des configurations spatiales particulières telles que l'ampglow ou les poussières.

Si nous essayons de quantifier l'intensité des pixels d'arrière-plan dans les différentes images dont nous disposons, nous pouvons écrire les expressions suivantes :

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

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

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

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

avec, L pour Lights, D pour Darks, F pour Flats et O pout Bias.

Pour les images L, la première partie est une composante d'illumination spatiale, *c'est-à-dire* $a - b(x - \frac{W}{2})^2$. Nous avons choisi ici une variation quadratique avec une valeur maximale a au milieu du cadre de largeur W, même autour du centre du capteur. Ce n'est pas la forme spatiale exacte du vignettage, mais c'est une assez bonne approximation pour comprendre comment il fonctionne. En plus de ce terme d'illumination spatiale, il existe un terme variant avec le temps d'exposition qui est généralement appelé courant d'obscurité ($d_{\text{rate}} \times t_{\text{lights}}$) mais qui ne dépend pas de la position du pixel sur le capteur. Enfin, il y a un pedestal, c'est-à-dire l'offset. Cet offset est présent dans toute image qui est prise, de sorte qu'on le retrouve dans toutes les expressions.

Les darks D n'étant pas éclairés, ils ne portent que le terme de courant d'obscurité, avec la même intensité que les images puisqu'ils sont tous les deux capturés avec la même durée, et le terme d'offset.

Les flats F ont également un terme spatial, proportionnel au terme trouvé dans les images "light". Le facteur K, plus grand que 1, indique simplement que leur intensité est plus grande. Pour écrire cela, il suffit de supposer que les pixels répondent linéairement au nombre de photons qu'ils recueillent, ce qui est raisonnable. Nous aurions également pu écrire un terme de courant d'obscurité, proportionnel au temps d'exposition des flats. Mais si ce temps n'est pas important, nous pouvons supposer qu'il est négligeable. Si ce n'est pas le cas, cela signifie qu'il faut prendre des darks de flat ou au moins évaluer leur niveau.

Et enfin l'offset O mesure seulement le niveau de l'offset.

Pour visualiser ces niveaux, nous avons représenté ci-dessous ces expressions sous forme de courbes en fonction de la position sur une image et nous vous encourageons à faire de même et à jouer avec les entrées.

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

Les valeurs L, D et O en ADU sont données sur l'échelle de gauche tandis que F est sur l'échelle reporté à droite.

Qu'est-ce que l'étalonnage des butes ? Lorsque vous étalonnez vos brutes, vous effectuez l'opération suivante :

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

Le terme $F - O$ désigne un flat dont vous avez soustrait le niveau d'offset (qu'il s'agisse d'un masterbias ou simplement d'un niveau). Il s'agit de l'opération effectuée avant l'empilement de votre masterflat. Et le terme $L - D$ représente une image dont vous avez soustrait le niveau du courant de dark et l'offset, c'est-à-dire un masterdark. Si vous remplacez par les expressions ci-dessus, vous obtenez ce qui suit :

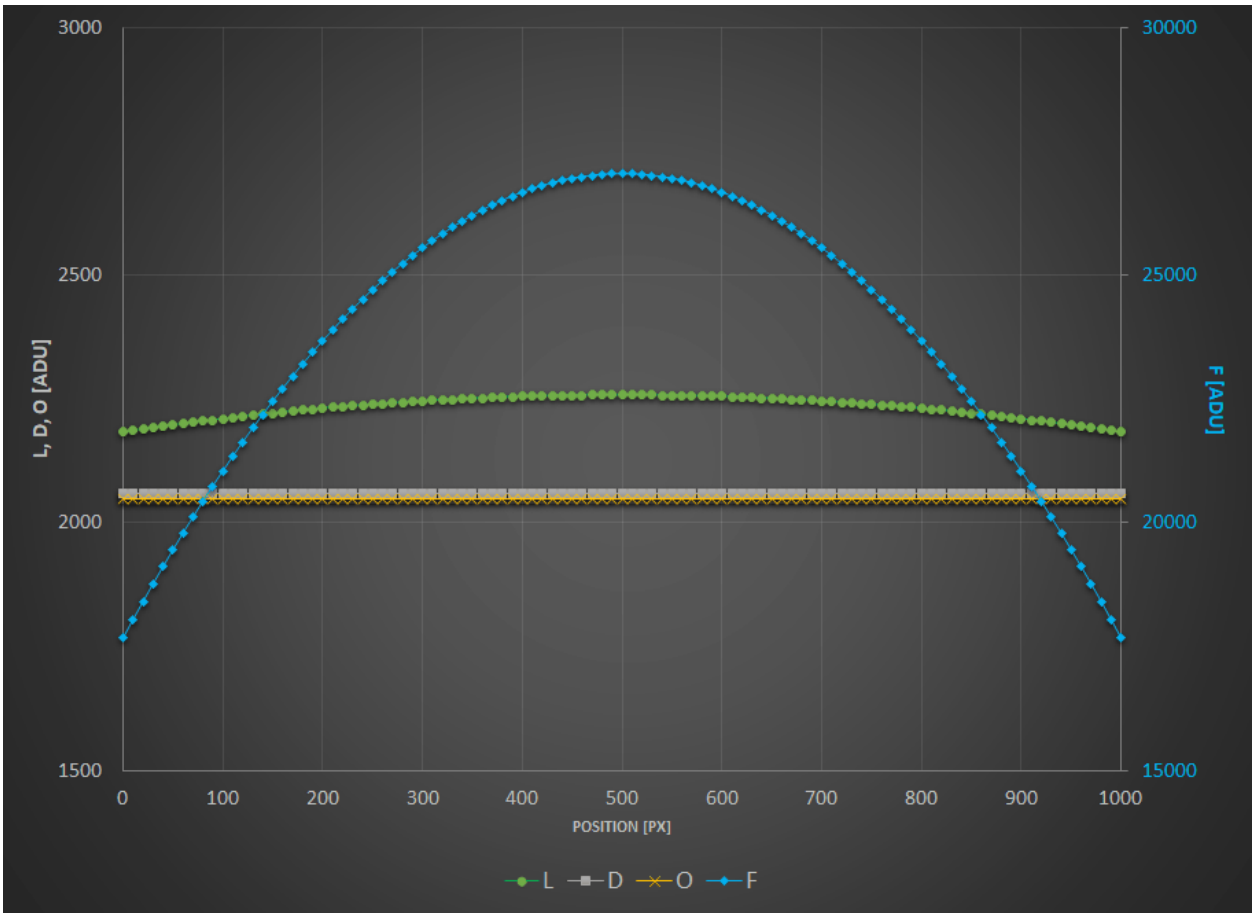
$$L_c = \frac{1}{K}.$$

Il n'y a plus de terme de variation spatiale, vous avez mis vos "aplati" vos images ! Obtenir une valeur raisonnable en ADU (et non pas $1/K$) est ce que Siril fait lorsque vous cochez *Evaluer automatiquement la valeur de normalisation* dans l'onglet *Calibration*.

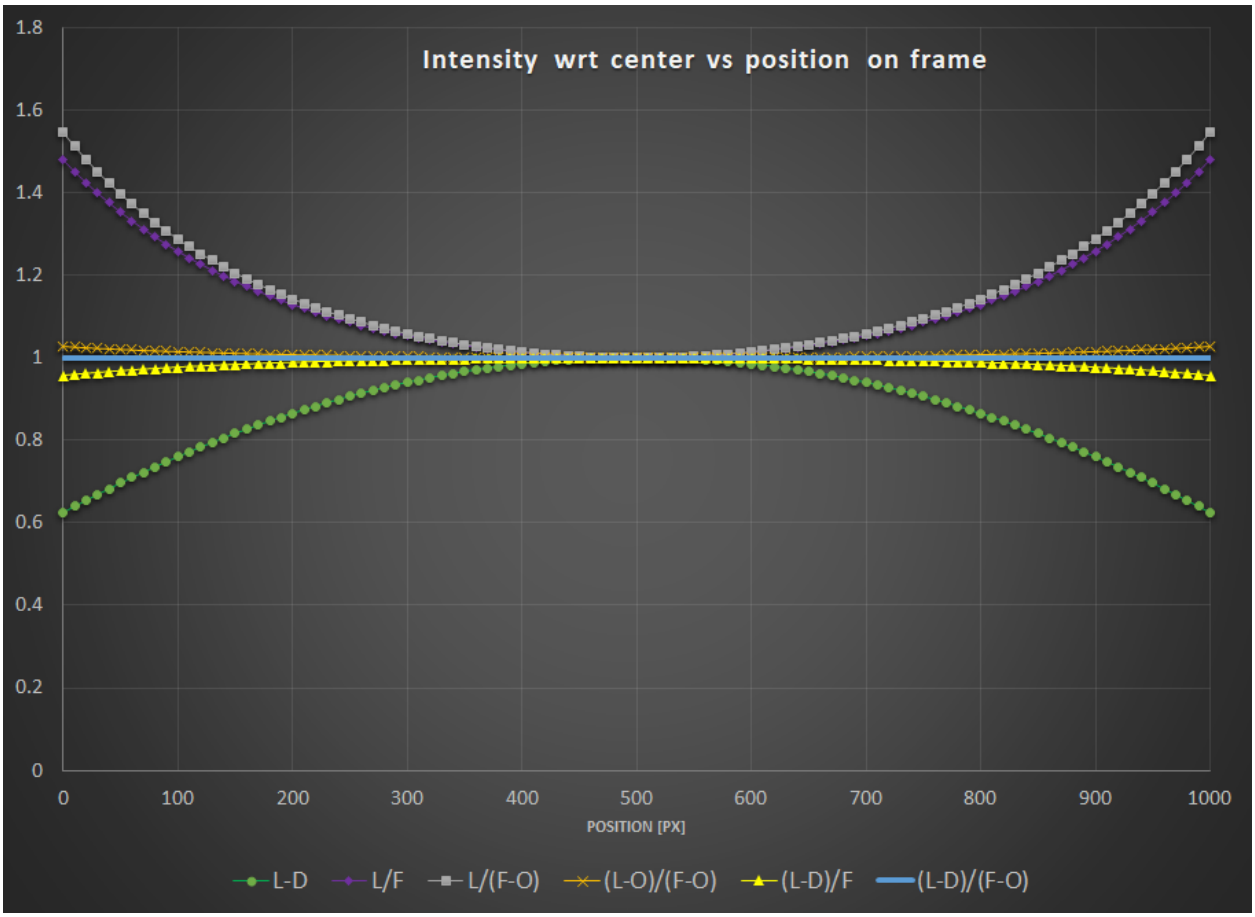
Et vous pouvez essayer avec n'importe quelle autre combinaison, aucune autre ne vous débarrassera des variations spatiales.

Pour illustrer cela, nous avons tracé ci-dessous le résultat de différentes combinaisons. Pour mettre tout sur la même échelle, tous les résultats sont normalisés pour avoir la même intensité de 1 au milieu du cadre. Les tests suivants sont présentés :

— $L - D$: vous avez réalisé que les darks.
— L/F : vous avez réalisé que les flats.
— $L/(F - O)$: vous avez juste réalisé des flats et les avez corrigé d'un offset (un maître ou un synthétique).
— $(L - O)/(F - O)$: vous avez juste des flats corrigé par un offset. Mais vous avez aussi soustrait l'offset des light.


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

— $(L-D)/F$: vous avez réalisé des flats et des darks mais pas d'offsets.
— $(L-D)/(F-O)$: vous avez tout fait dans les règles de l'art.



Il est intéressant de noter que :
— $L-D$ ne montre évidemment aucune correction pour le vignettage.
— Les images L/F et $L/(F-O)$ présentent une sur-correction ou un vignettage inverse.
— Pour se rapprocher du résultat optimal, $(L-D)/F$ et $(L-O)/(F-O)$ montrent un champ presque plat. Cela dépend bien sûr de l'intensité du courant d'obscurité de votre capteur et du vignettage de votre train optique.
— La calibration de référence donne un champ plat.

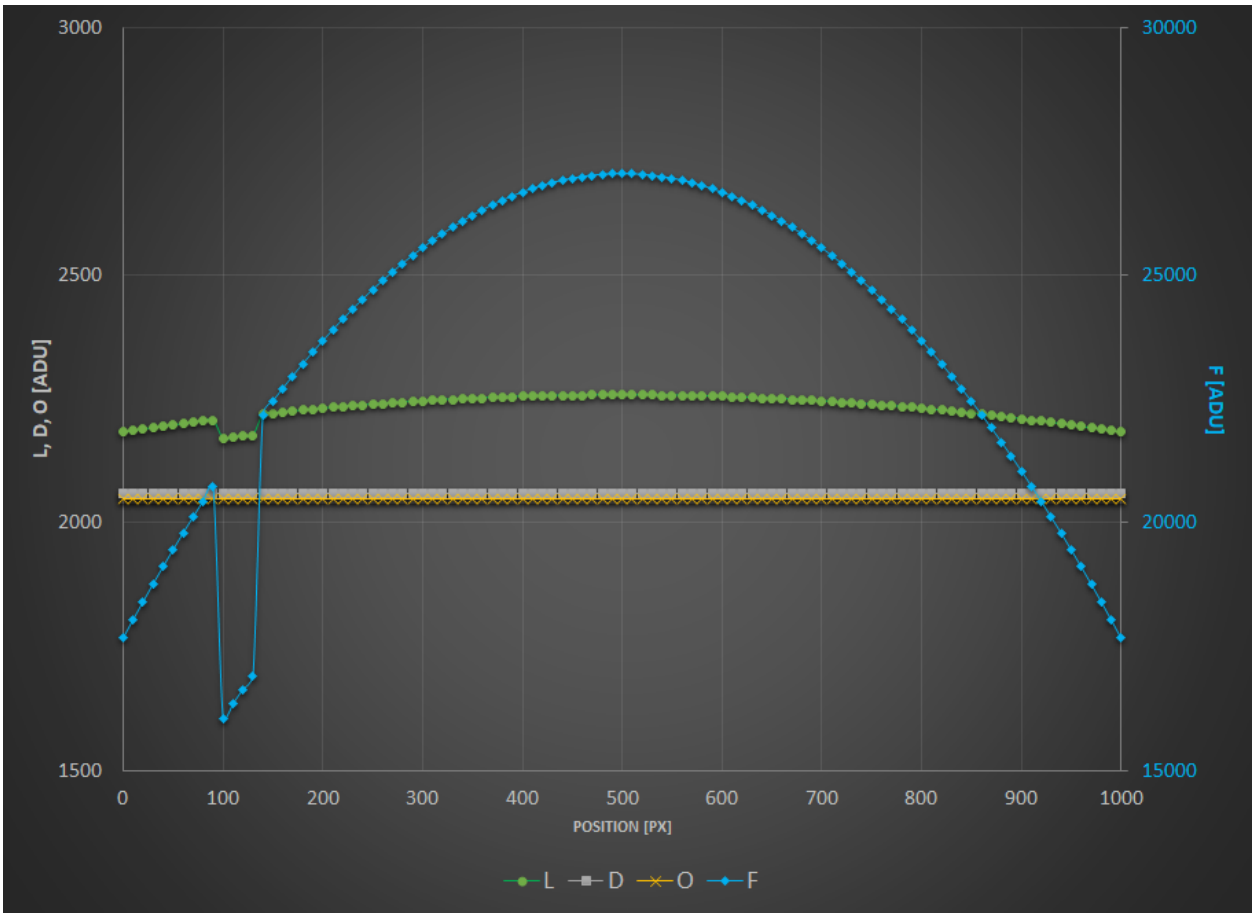
Les conclusions que nous pouvons tirer de ce qui précède sont les suivantes :
— Il est préférable de corriger vos lights avec un offset (masterbias ou simplement un niveau) si vous n'avez pas réalisé de darks.
— Mieux encore, si vous n'avez pas le temps de prendre une série de darks, cela vaut probablement la peine de prendre au moins un dark, de mesurer sa médiane, et de soustraire ce dark (synthétique) de vos lights. Bien sûr, cela ne corrigera pas l'ampglow ou ne permettra pas de corriger les pixels chauds, mais vos lights seront au moins plats !

Maintenant, qu'en est-il de la poussière... ?

Pour que vos flats corrigent aussi ces vilaines taches, la triste nouvelle est que vous devez aussi inclure toutes les images d'étalonnage dans l'équation. Nous avons ajouté un petit déficit ADU local dans les lights et les flats pour illustrer cet effet.

Comme vous pouvez le constater, seule la combinaison $(L-D)/(F-O)$ permet de s'en débarrasser.

Pour mieux illustrer les équations et les courbes ci-dessus, rien ne vaut un exemple concret. Toutes les photos ci-dessous sont reproduites avec l'aimable autorisation de G. Attard.


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

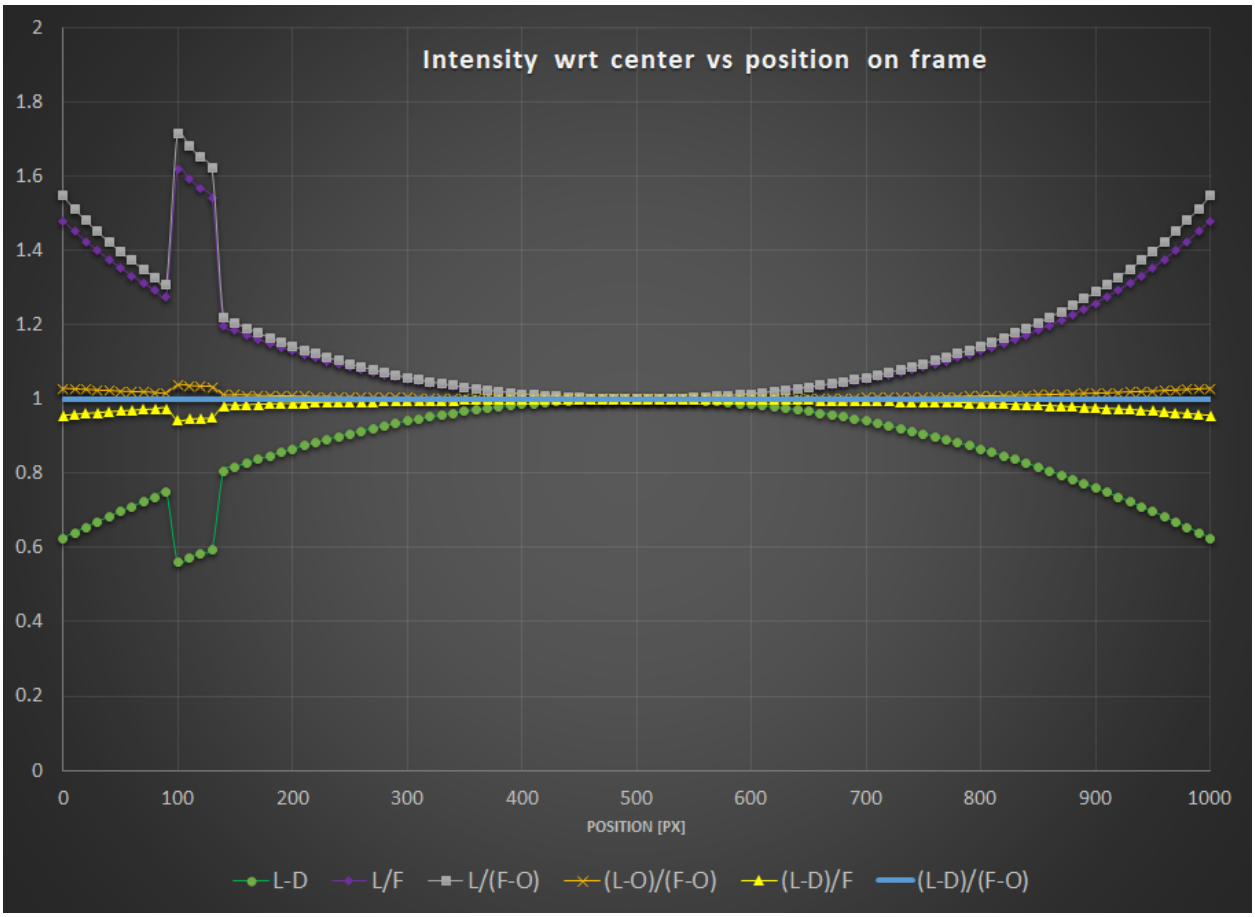

Intensity wrt center vs position on frame
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000
POSITION [PX]
L-D
L/F
L/(F-O)
(L-O)/(F-O)
(L-D)/F
(L-D)/(F-O)

FIG. 15 – $L - D$

FIG. 16 – L/F

FIG. 17 – $L/(F-O)$

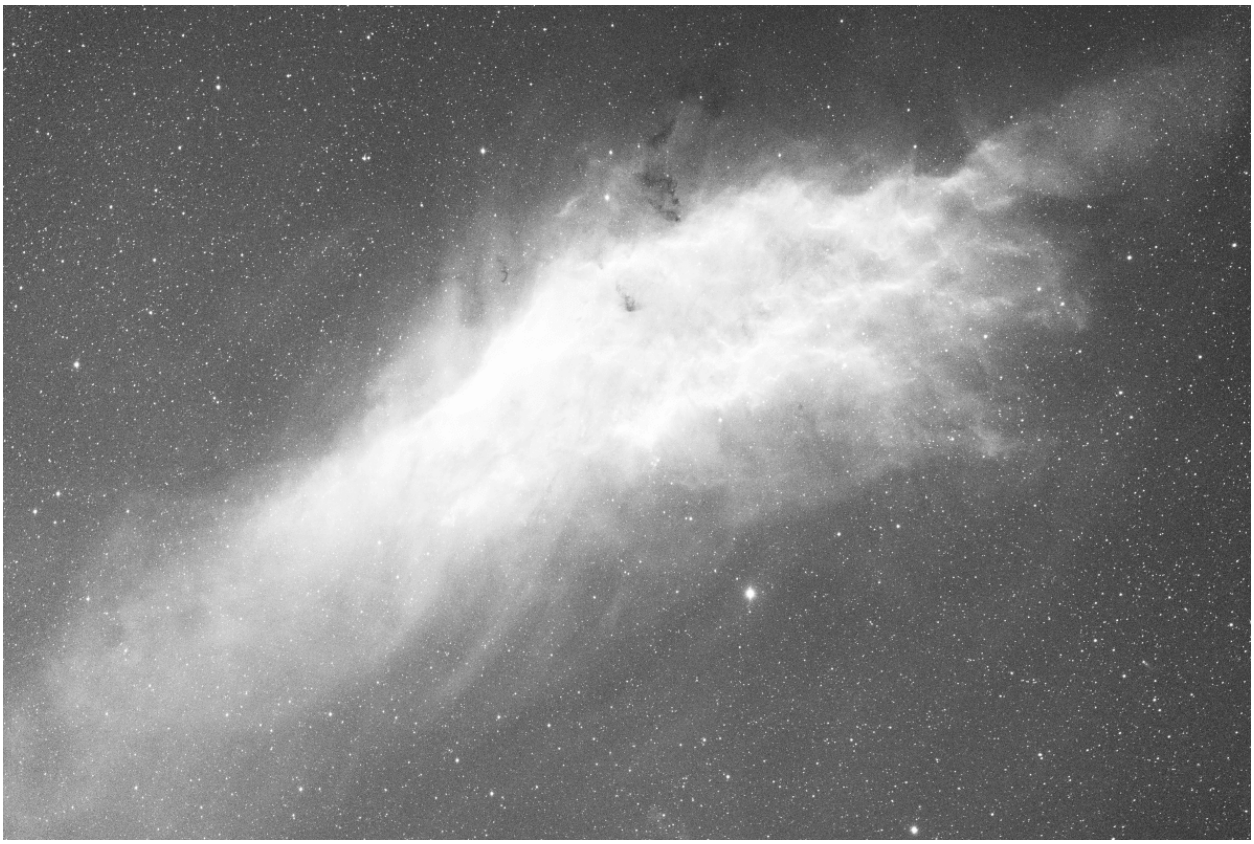


FIG. 18 – $(L - O)/(F - O)$

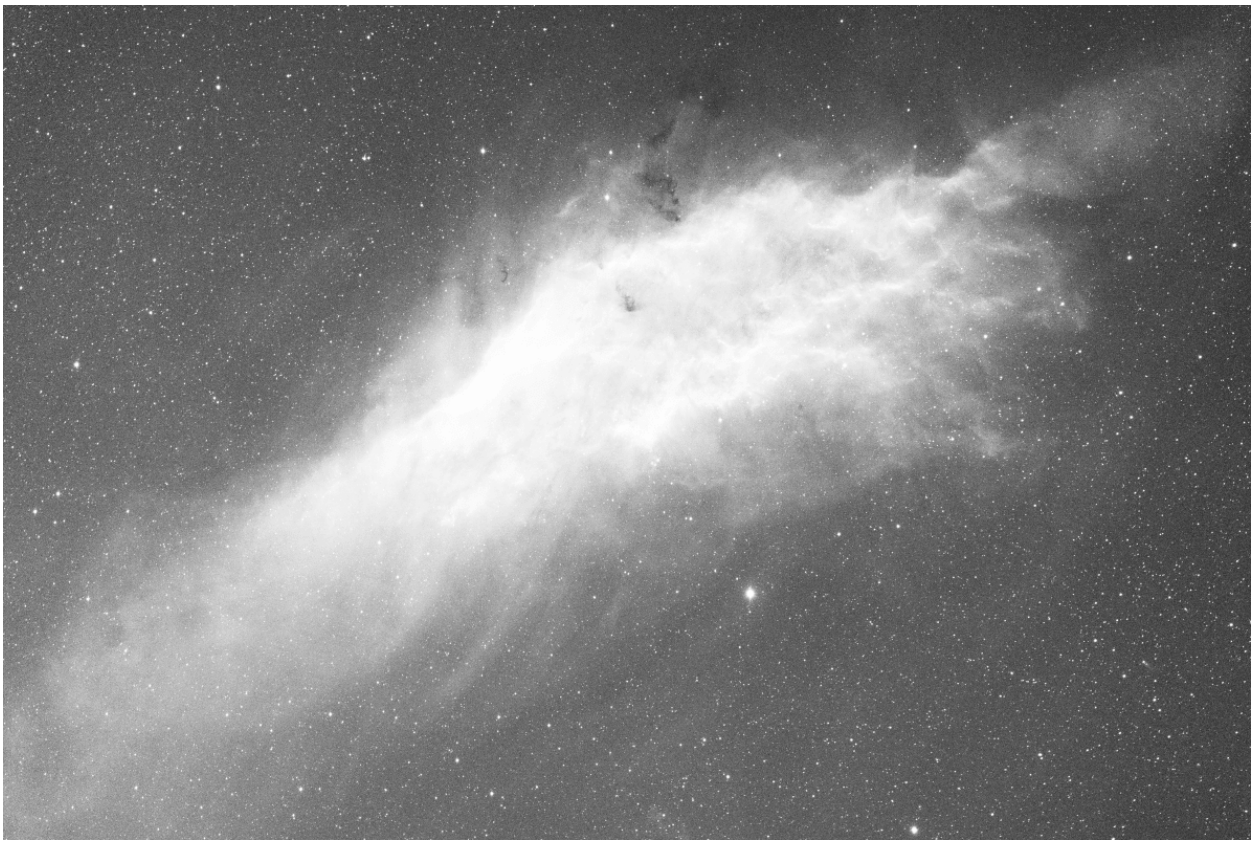


FIG. 19 – $(L - D)/(F - O)$

9.2.4 Résolution des problèmes de calibration

La calibration est une étape arithmétiquement très simple, qui ne peut échouer si les données d'entrée sont conformes à ce que l'on attend des images astronomiques.

Cependant, les utilisateurs sont régulièrement confrontés à des situations où les images calibrées ne sont pas correctes. Dans cette section, nous vous donnerons un aperçu des problèmes rencontrés et de la manière de les éviter.

Tout d'abord, l'outil *statistique* est une aide précieuse pour comprendre les problèmes, et est utilisé dans la majorité des cas pour les résoudre.

- Lors de l'analyse des statistiques d'un dark maître, celui-ci doit d'abord être noir. En effet, ces images sont prises avec le bouchon fermé et il n'y a pas de raison que l'un des photosites soit privilégié. L'image doit donner l'impression d'avoir été prise par un capteur monochrome, la matrice de Bayer n'étant alors pas visible. Ci-dessous, voici un exemple où le master-dark a subi une balance des couleurs non souhaitée pour ce type d'image. En conséquence, il n'est plus noir et la matrice de Bayer est visible. Un tel dark est impropre à l'utilisation et doit être refait.
- Pendant la session de nuit, il est très important de régler la valeur OFFSET sur la même valeur pour toutes les images. En particulier, il est obligatoire d'avoir le même réglage pour les paires darks/lights et bias/flats. Si la première condition n'est pas respectée, des données importantes peuvent être perdues (écrêtage des pixels sur le côté gauche de l'histogramme). Si les deux conditions ne sont pas remplies, il est très probable que vos images ne pourront pas être correctement traitées avec le flat (voir la *section ci-dessus*).
- Vérifier les niveaux des darks et des images : la valeur médiane des images doit être suffisamment élevée par rapport à celle du master-dark pour éviter de générer des images pleines de pixels avec des valeurs négatives.
- Si vous avez utilisé les mêmes réglages pour les darks et bias, leurs valeurs médianes devraient être très proches les unes des autres (du moins avec une caméra refroidie). Dans le cas contraire, cela peut signifier que vous avez une fuite de lumière qui a affecté vos darks (les bias sont moins sensibles car ils sont exposées pendant un temps beaucoup plus court). Inspectez donc toujours votre master-dark pour voir s'il y a un gradient ou une zone plus claire au centre. Ce phénomène ne doit pas être confondu avec l'ampglow qui est normal pour certaines caméras.
- Nous vous recommandons vivement de réaliser vos images de la même manière : même logiciel / même ordinateur ou astrobox / même format d'image. En effet, chaque logiciel peut utiliser ses propres conventions d'écriture, et les images peuvent ne plus être compatibles entre elles. On entend souvent parler d'utilisateurs qui font toutes leurs images avec une astrobox, et qui font les flats le lendemain directement avec leur reflex numérique. Dans ce cas, les images sont souvent de tailles différentes, ce qui rend la calibration impossible.
- Une erreur souvent rencontrée lors de l'exécution d'un script est la présence d'images JPG dans l'un des dossiers d'entrée (darks/biases/flats/lights), le plus souvent des instantanés sauvegardés par le logiciel d'acquisition pour une navigation plus rapide. La conséquence de ce type d'erreur est que la calibration échoue et s'arrête, se plaignant que les images ne sont pas de la même taille. En fait, les images JPG étant déjà dématricées, elles possèdent trois canaux, alors que les images RAW n'en ont qu'un. Supprimez toutes les images JPG des dossiers d'entrée pour résoudre ce problème.
- Vérifiez que le flat n'est pas surexposé. Les flats sont utilisés pour corriger les variations de sensibilité d'un pixel à l'autre dans le capteur. Si certains pixels sont surexposés, leur sensibilité réelle risque de ne pas être représentée avec précision, ce qui entraînera des corrections incorrectes au cours du processus d'étalonnage. Un flat surexposé est la garantie d'une calibration ratée.
  Pour vérifier la présence de pixels surexposés, vous pouvez charger un flat et utiliser l'option *Traitement de l'image → Transformation de l'histogramme...* pour afficher l'histogramme de l'image. L'image ci-dessous montre que l'un des pics est écrêté à droite. Par prudence, vous devriez toujours vérifier que la queue droite du pic le plus à droite n'est pas supérieure à 80 %, afin d'éviter d'entrer dans une zone où votre capteur pourrait devenir non linéaire.

9.3 Alignement

La registration ou alignement est essentiellement le processus d'alignement des images d'une séquence pour pouvoir les traiter par la suite. Tous les processus décrits ci-après calculent la transformation à appliquer à chaque image afin d'être alignée avec l'image de référence de la séquence.

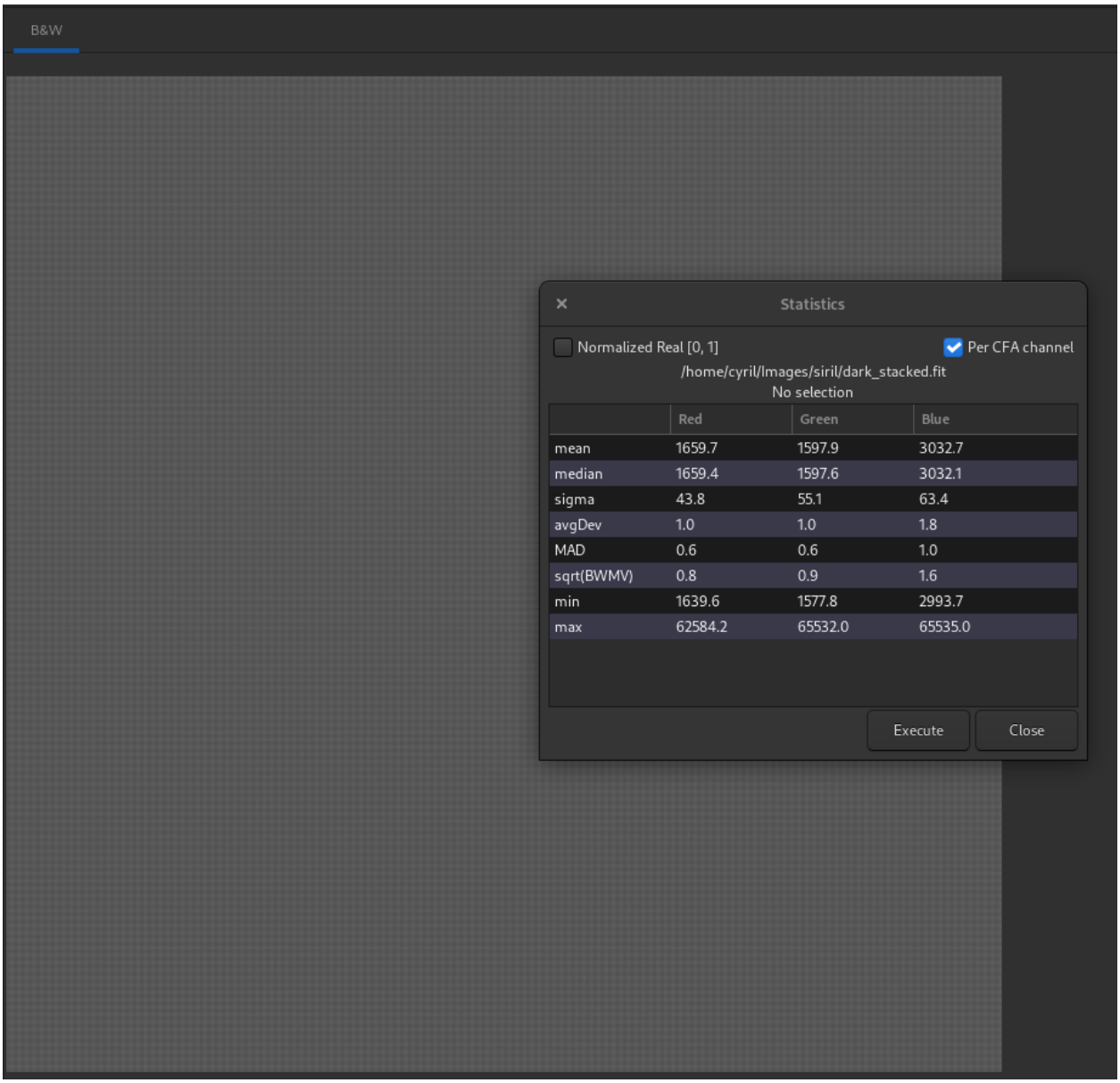


FIG. 20 – Un examen attentif des statistiques montre que la valeur médiane de chaque canal est différente, alors qu'elle devrait être identique (ou presque). En outre, le motif de Bayer est clairement visible.

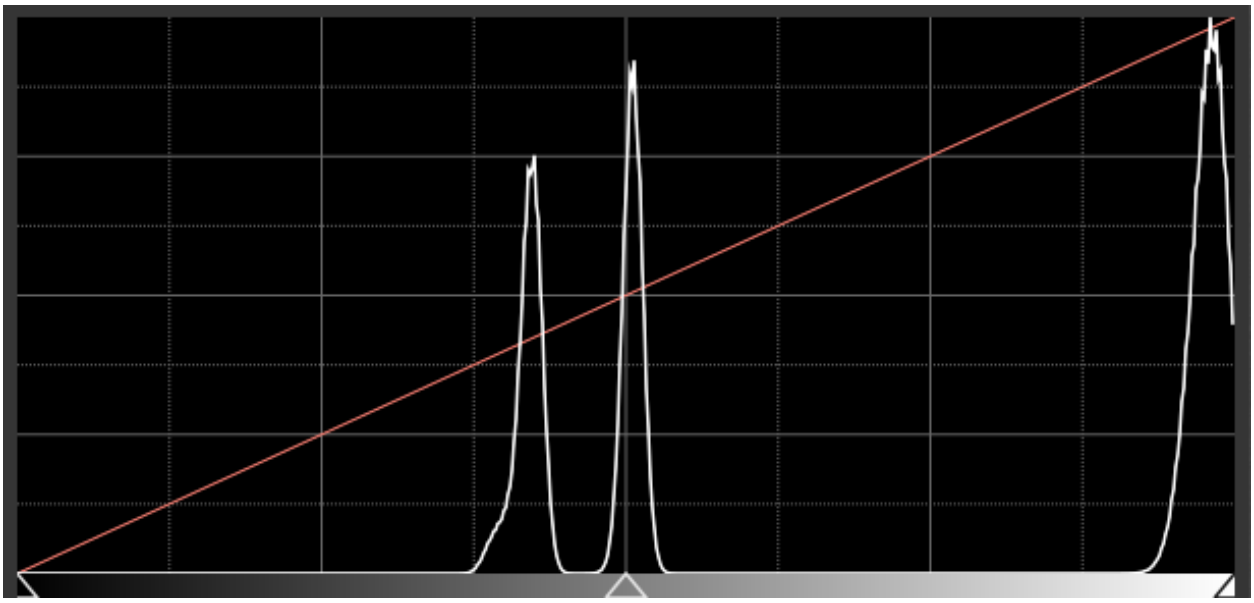


FIG. 21 – Saturation d'un flat. Lorsque cela se produit, cela signifie qu'il faut réduire le gain ou le temps d'exposition.

La force de Siril réside dans la grande variété d'algorithmes d'alignement proposés. Chaque méthode est expliquée ci-dessous. L'appui sur le bouton *Aligner* lance l'alignement de la séquence.

Il est possible de choisir le **canal d'alignement**. Le vert est le canal par défaut pour les images en couleur, la luminance pour les monochromes. Le signe (*) apparaissant après le nom du canal signifie que des données d'alignement sont déjà disponibles pour ce canal. Lors du traitement des images, les données d'alignement sont extraites du canal par défaut s'il est disponible (pour les images RVB : Vert, sinon retour au Bleu puis au Rouge).

9.3.1 Théorie

Processus d'alignement

Ce que nous appelons Alignement est en fait un processus en trois étapes :

1. Détecter les caractéristiques à faire correspondre dans toutes les images
2. Calculer les transformations entre chaque image et l'image de référence
3. Appliquer la transformation calculée à chaque image pour obtenir de nouvelles images

Selon la méthode d'alignement choisie, les 3 étapes se déroulent (ou non) en un seul processus. Siril utilise les valeurs par défaut les plus raisonnables (choisir d'appliquer ou non la transformation calculée) en fonction de la méthode d'alignement sélectionnée, mais la compréhension de la machinerie interne peut vous aider à modifier ce comportement pour mieux répondre à vos besoins.

Algorithmes

Le tableau ci-dessous détaille les différents algorithmes utilisés pour les 2 premières étapes (détection et calcul de la transformation).

Méthode d'alignement	Détection des caractéristiques	Calcul de la transformation	Nouvelle séquence
Global	*PSF Dynamique*	Correspondance des triangles + RANSAC	Y
Global avec 2 passe			N
1-2-3 étoiles	PSF *minimisation* dans la boîte de sélection	Décomposition en valeur singulière (2-3 étoiles) Différence (1 étoile)	N
Alignement par motif de l'image	correlation croisée avec boîte de sélection		N
KOMBAT	Max de convolution dans le domaine spatial dans la boîte de sélection		N
Comète	PSF *minimisation* dans la boîte de sélection	Décalages du vecteur vitesse à l'aide d'horodatages	Y(*)
Manuelle	Vos yeux	Vos mains	N
Astrométrie	*PSF Dynamique*	À partir de la solution astrométrique	N

Note

(*) Comet registration creates a new sequence file (with the prefix *comet_* by default) and creates "new" images. If registration data was present in the input sequence, the comet shifts are composed with this pre-existing registration data. The "new" images created are simply symlinks to the input images, so that they do not use up additional storage space. This change introduced in version 1.4 was made to clarify whether or not a sequence containing registration data, had only star alignment or star+comet shift alignment. Exporting a new sequence now makes it obvious that additional shifts are included. It also enables to use the same input sequence and generating multiple sequences aligned on different moving objects.

The table below lists the transformations compatible with each method, and if undistortion is available as well.

Méthode d'alignement	Translation	Euclidien	Similarité	Affine	Homographie	Annuler la distorsion
Global	sous-pixel		x	x	x	x
2 passes	sous-pixel		x	x	x	x
1-2-3 étoiles	sous-pixel (1-2-3)	(2-3)				
Alignement par motif de l'image	pixel					
KOMBAT	pixel					
Comète	sous-pixel					
Manuelle	pixel					
Astrométrie					x	x

Il est également important de garder à l'esprit comment la séquence alignée est introduite dans le processus d'empilement qui est généralement utilisé juste après l'alignement :

— si la transformation consiste uniquement en des décalages de pixels, l'algorithme d'empilement peut utiliser ces décalages à la volée lors de la lecture des images. Cela signifie qu'il n'est pas nécessaire de générer des "images alignées". Cela permet de gagner de l'espace de stockage et d'éviter l'interpolation. C'est, bien sûr, au détriment d'un alignement moins précis (c'est-à-dire d'une précision inférieure au pixel) mais c'est généralement utilisé sur des images planétaires/de chance où l'échantillonnage est faible. Cette méthode peut également être appliquée avec une méthode d'alignement qui calcule les décalages sous-pixel. Au cours du processus d'empilement, les décalages seront arrondis à la précision du pixel. Dans tout autre cas, c'est-à-dire lorsque l'empilement est alimenté par une séquence où l'alignement a calculé des transformations plus complexes que de simples décalages

mais que les images alignées n'ont pas été sauvegardées, Siril émettra un avertissement vous invitant à exporter les images alignées avant de procéder à l'empilement.
— Dans tous les autres cas, une fois les transformations calculées, les images transformées doivent être sauvegardées avant de procéder à l'empilement, généralement nommées avec le préfixe `r_`.

Transformations de l'image

Transformation linéaire

Siril utilise des transformations linéaires, avec différents degrés de liberté, pour mettre en correspondance une image avec l'image de référence :

— **Translation** est un mapping rigide à 2 degrés de liberté (décalages x/y), bien adapté aux images sans distorsion, sans mise à l'échelle et sans rotation de champ. Il ne nécessite qu'une seule paire d'étoiles (ou de caractéristiques) à faire correspondre pour définir la transformation.
— **Euclidien** est un mapping rigide à 3 degrés de liberté (déplacements x/y + une rotation), pour des images sans distorsion, sans mise à l'échelle. Il faut qu'au moins 2 paires d'étoiles soient appariées pour définir la transformation.
— **Similarité** est un mapping à 4 degrés de liberté (une échelle, une rotation et des décalages x/y) plus rigide que l'homographie, bien adaptée aux images sans distorsion. Il faut qu'au moins 2 paires d'étoiles soient appariées pour définir la transformation.
— **Affine** est un mapping à 6 degrés de liberté (deux échelles, un cisaillement, une rotation et des décalages x/y) plus rigide que l'homographie, bien adapté aux images avec peu de distorsion. Elle nécessite la mise en correspondance d'au moins 3 paires d'étoiles pour définir la transformation.
— **Homographie** est la transformation par défaut qui utilise une transformation à 8 degrés de liberté pour déformer les images sur le cadre de référence. Cette transformation est bien adaptée au cas général et fortement recommandée pour les images à grand champ. Elle nécessite la mise en correspondance d'au moins 4 paires d'étoiles pour définir la transformation.

Distorsions

Since version 1.3, Siril can also account for distortions, for some of the registrations methods as listed in this *table*. Distortion coefficients handled by Siril follow SIP convention. This convention assumes that the pixel coordinates need to be corrected BEFORE trying to map them through a linear transformation. In WCS jargon, this is called a *prior* distortion (as opposed to a *sequent* distortion).

Those coefficients are used twice during the registration process :

— the stars positions detected in "distorted" conditions are first corrected, both in the image to be aligned and the reference image (or the stars projected positions for astrometric registration). The *linear transformation* that maps the stars from the current image to the reference is then computed.
— When exporting the registered image, it is first corrected for distortion and then linearly projected to be aligned to the reference image. Note that this actually occurs in a single operation (the pixel mapping is computed as the composition of this non-linear correction and then the linear projection) so as to avoid interpolating pixel values twice. The reference image also undergoes this correction without the linear preojection.

Image de référence

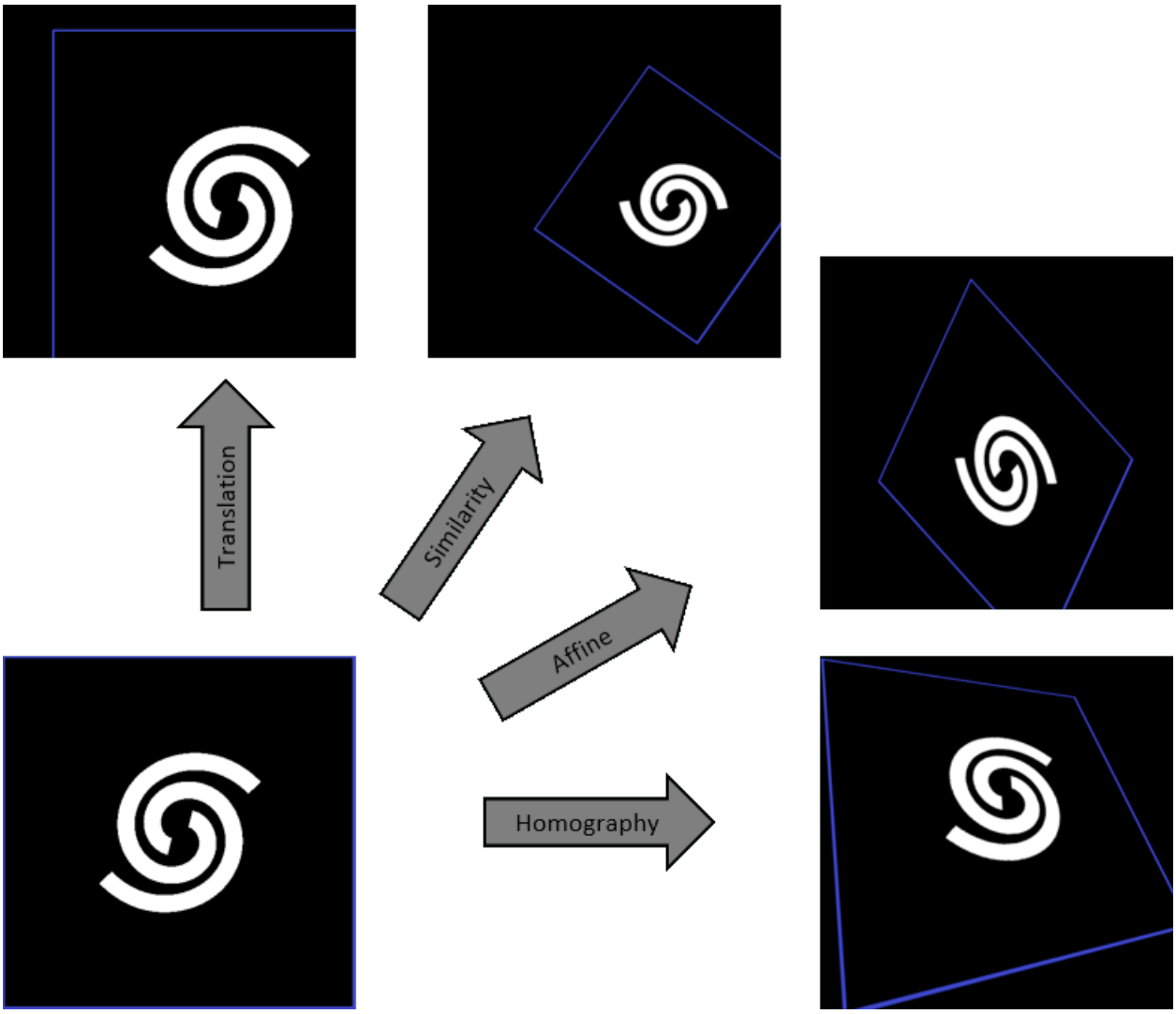
C'est l'image qui est utilisée comme référence commune pour calculer les transformations qui envoient toutes les images de la séquence sur cette image particulière.

Si elle n'est pas définie manuellement, l'image de référence est choisie selon les critères suivants :

— si la séquence a déjà été alignée, il s'agit de la meilleure image, en termes de FWHM la plus faible ou de qualité la plus élevée selon le type d'alignement
— Sinon, c'est la première image de la séquence qui n'est pas exclue.

Pour spécifier une image comme référence, vous pouvez :

— Ouvrir le *sélecteur d'images*, sélectionner l'image à définir comme nouvelle référence et cliquer sur le bouton *Image de référence*.


Translation
Similarity
Affine
Homography

— Utiliser la commande *setref*. Par exemple, si vous voulez définir l'image n°10 comme référence :

```
setref 10
```

Ligne de commande Siril

```
setref sequencename image_number
```

Définit l'image de référence de la séquence donnée en premier argument. **image_number** est le numéro séquentiel de l'image dans la séquence, et non le numéro dans le nom de fichier, commençant à 1

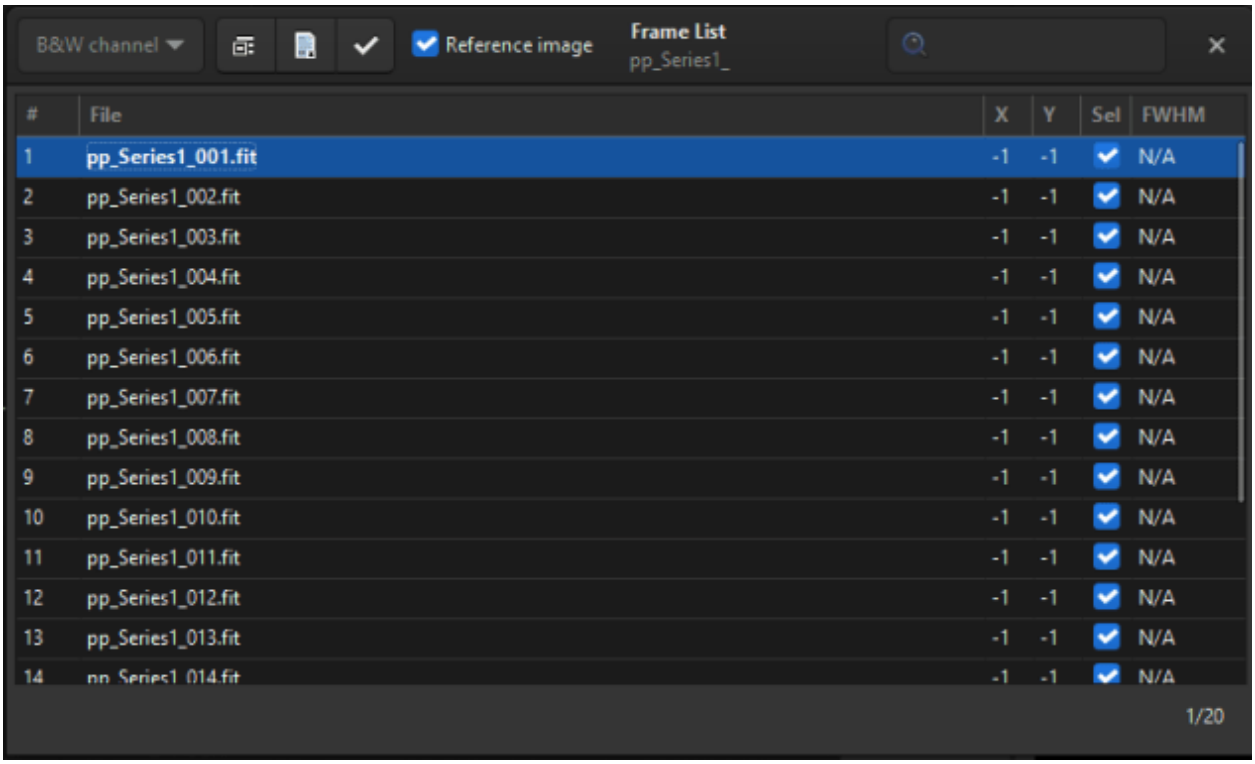


FIG. 22 – La boîte de dialogue de la liste des images. Vous pouvez parcourir toutes les images de la séquence.

Pendant l'empilement, l'image de référence est également utilisée comme référence de normalisation, si la normalisation est activée.

9.3.2 Méthodes d'alignement

Alignement global

This is the preferred algorithm to align deep-sky images with sufficient overlap.

L'alignement global est basé sur la méthode de similarité des triangles pour identifier automatiquement les étoiles communes dans chaque image [Valdes1995]. Notre implémentation est basée sur le programme match de Michael Richmond. Ensuite, l'algorithme RANSAC [Fischler1981] est utilisé sur les listes d'étoiles pour rejeter les aberrations et déterminer la matrice de projection. La robustesse de l'algorithme dépend de la capacité à détecter les étoiles tout en évitant les fausses détections. Siril dispose d'un algorithme de détection des étoiles très élaboré qui évite autant que possible de sélectionner des objets qui ne sont pas des étoiles dans le temps le plus court possible. La détection

des étoiles les plus brillantes est généralement la plus importante. Cependant, s'il est nécessaire de détecter des étoiles moins brillantes, alors la fenêtre *PSF Dynamique* peut être utilisée pour ajuster les paramètres de détection.

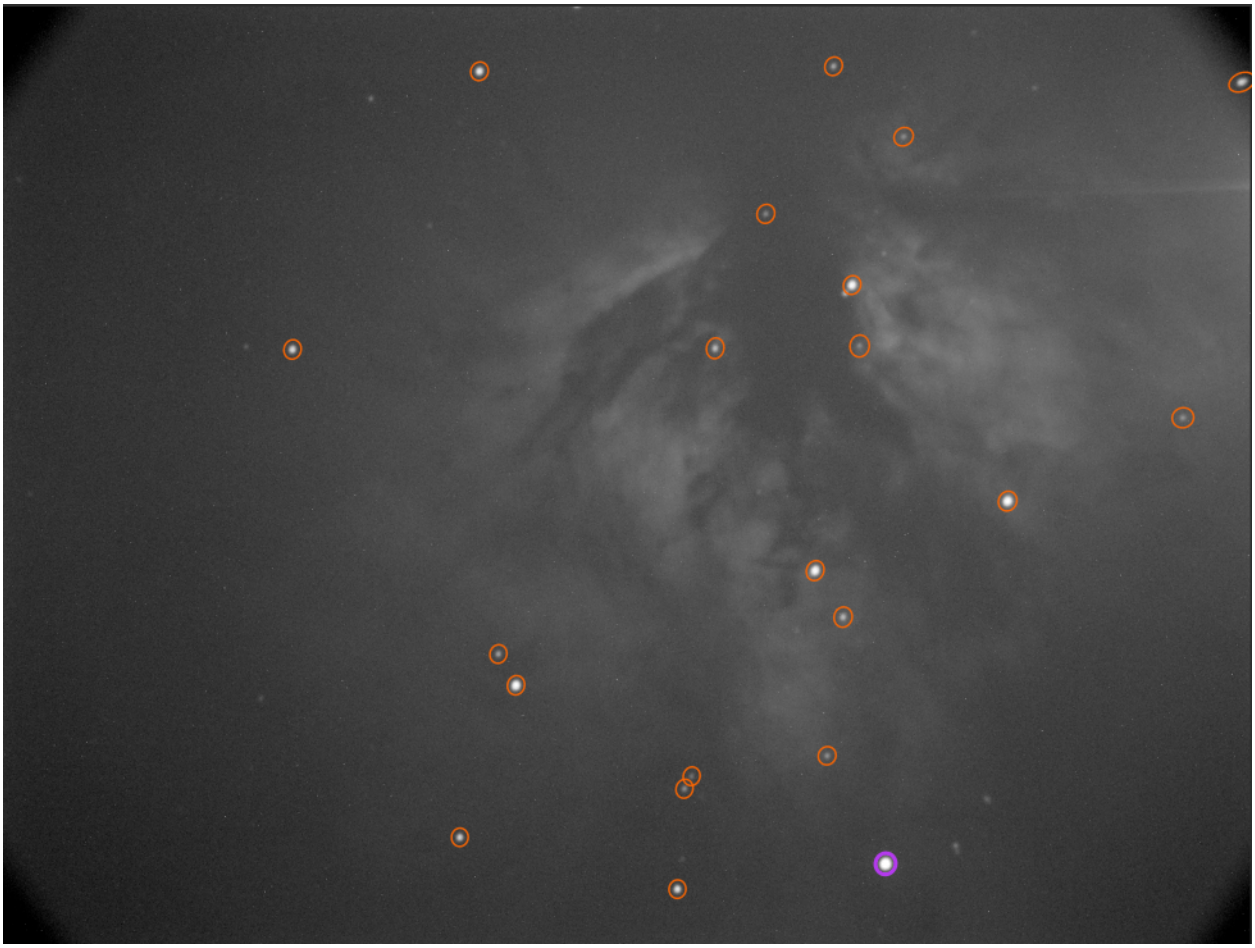


Fig. 23 – Détection automatique des étoiles dans une image unique

There are some options associated with this alignment method.

Le menu déroulant *Transformation* permet de choisir entre différentes transformations.

Avertissement

La mise en correspondance initiale des étoiles utilise l'algorithme de similarité des triangles, en conséquence le minimum de paires d'étoiles doit être au moins de 3 pour **Translation**, **Similarité** et **Affine** et de 4 pour **Homographie**.

Les autres options d'alignement sont :

— 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 Dynamic PSF tool.

— L'option *Maximum d'étoiles ajustées* définit le nombre maximum d'étoiles à rechercher dans chaque image (par défaut 2000). Plus cette valeur est grande, plus il y aura d'étoiles potentiellement détectées, résultant en une détection plus longue mais un alignement plus précis.

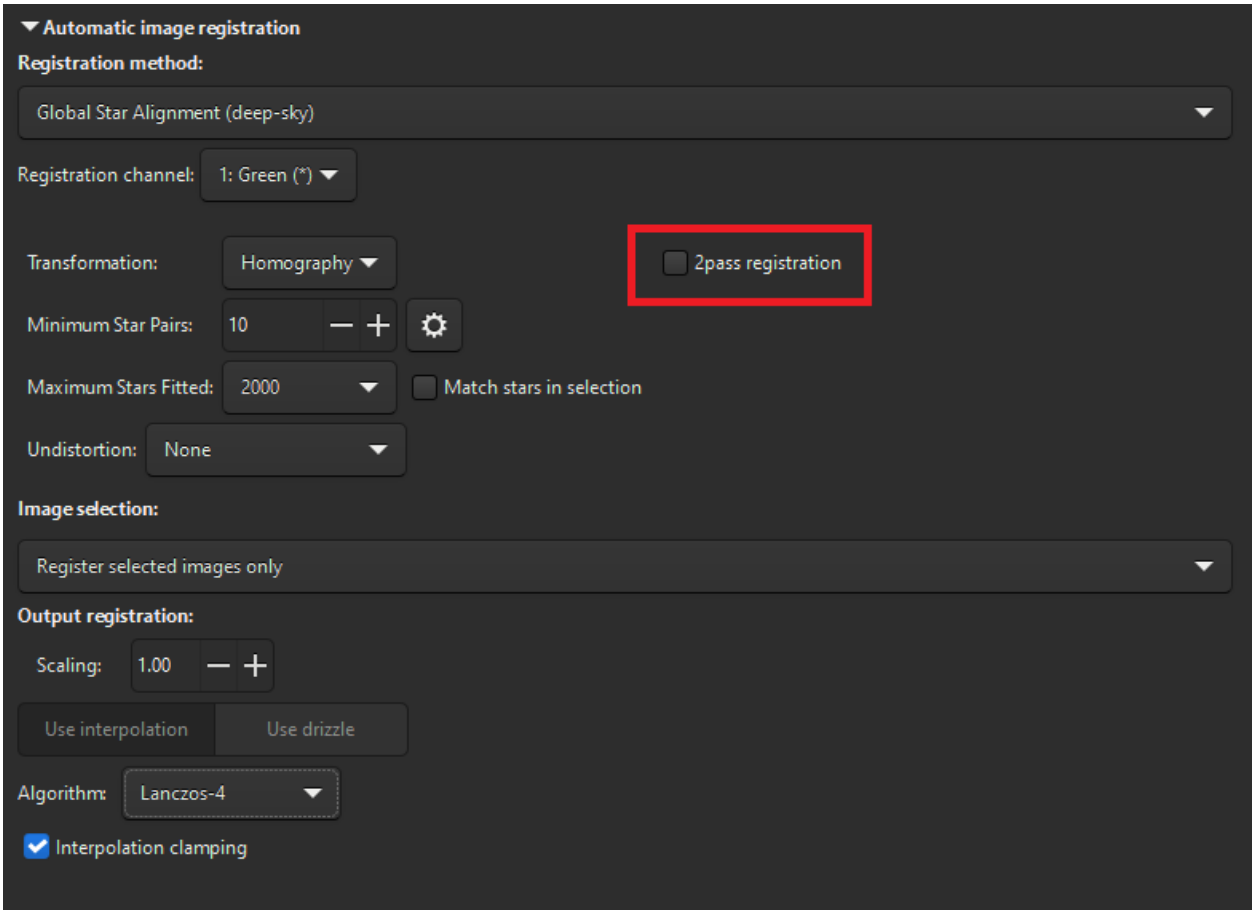


FIG. 24 – Options d'alignement global

— Use the 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 is done, this option is ignored.
— The *Undistortion* dropdown menu allows to choose between different corrections :
  — Aucun
  — À partir de l'image
  — À partir d'un fichier FITS/WCS
  — A partir des maîtres

From image uses the astrometric solution from the currently loaded image.
From FITS/WCS file will open a file chooser to select a FITS image or a `*.wcs` file that has distortion coefficients.
From masters will fetch the master distortion file corresponding to each image. This option can be useful if you are aligning images from two or more sessions and the sensor has been moved wrt. the optics, in which case distortion coefficients may not be the same.

Astuce

For this last option *From masters* to work, you will need to have set a master distortion path in the *preferences*. To create a master file, go *here*.

The options at the bottom let you :
— Éliminez les images qui ont été désélectionnées de la séquence.
— Choose between interpolation and drizzle for exporting the images. Those are the same as in the *Output registration* section, they are not further explained here.

Ligne de commande Siril

```
register sequencename [-2pass] [-selected] [-prefix=] [-scale=]
register sequencename ... [-layer=] [-transf=] [-minpairs=] [-maxstars=] [-
↪nostarlist] [-disto=]
register sequencename ... [-interp=] [-noclamp]
register sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
```

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.

Toutes les images de la séquence seront alignées sauf si l'option **-selected** est passée, dans ce cas les images exclues ne seront pas traitées.
The **-2pass** option 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, based on image quality and framing. To generate transformed images after this pass, use SEQAPPLYREG.
If created, the output sequence name will start with the prefix "r_" unless otherwise specified with **-prefix=** option. The output images can be rescaled by passing a **-scale=** argument with a float value between 0.1 and 3.

Options de transformation d'image :

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.

-transf= specifies the use of either **shift**, **similarity**, **affine** or **homography** (default) transformations respectively.

-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.

-maxstars= will specify the maximum number of stars 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.

-nostarlist désactive la sauvegarde de la liste d'étoiles sur le disque.

-disto= uses distortion terms from a previous platesolve solution (with a SIP order > 1). It takes as parameter either **image** to use the solution contained in the currently loaded image, **file** followed by the path to the image containing the solution or **master** to load automatically the matching distortion master corresponding to each image. When using this option, the polynomials are used both to correct star positions before computing the transformation and to undistort the images when output images are exported.

Option d'interpolation de l'image :

Par défaut, les transformations sont appliquées pour aligner les images à l'aide de l'interpolation.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.

La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**.

Options de drizzle pour les images :

Sinon, les images peuvent être exportées à l'aide de l'algorithme HST drizzle en passant l'argument **-drizzle** qui peut prendre les options supplémentaires suivantes :

-pixfrac= défini la fraction de pixel (par défaut =1.0).

L'argument **-kernel=** défini le noyau de drizzle et doit être suivi par l'un des arguments suivant : **point**, **turbo**, **square**, **gaussian**, **lanczos2** or **lanczos3**. La valeur par défaut est **square**.

L'argument **-flat=** spécifie un flat maître pour pondérer les pixels d'entrée drizzlés (la valeur par défaut est sans flat).

Remarque : lorsque vous utilisez **-drizzle** sur des images prises avec un capteur couleur, les images d'entrée ne doivent pas être débayerisées. Dans ce cas, la détection des étoiles se fera toujours sur les pixels verts

Liens : *setfindstar*, *psf*, *seqapplyreg*

Alignement 2 passes

To activate this option, simply tick the *corresponding box* after choosing *Global registration* in the method drop-down.

This performs only the first (of 2 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 Registation* algorithm but this method does not create any sequences and all alignment information is saved in the `seq` file.

Pendant la détection des étoiles, Siril fixe un maximum de 2000 étoiles à trouver (ceci peut aussi être changé avec l'option appropriée). Dans le cas où plus d'une image a atteint les limites maximales d'étoiles, les listes d'étoiles de toutes les images sont examinées à nouveau. Un nouveau seuil de détection minimum est défini pour pouvoir trier les images à la fois par nombre d'étoiles détectées et par FWHM.

The star lists of all images are saved, 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.

Cet alignement doit généralement être suivi par *Appliquer l'alignement existant* afin d'appliquer la transformation et de construire une nouvelle séquence, à moins que vous n'ayez choisi de calculer le `Translation`.

Ces lignes effectuent un alignement en deux passes sur une séquence nommée *pp_light* et l'appliquent. La sortie est une séquence *pp_light*.

```
# Align lights in 2 passes
register pp_light -2pass
seqapplyreg pp_light
```

Ces lignes effectuent un alignement en deux passes sur une séquence nommée *colors* et l'appliquent tout en recadrant les images de sortie à la surface commune minimale. La sortie est une séquence *pp_colors*. Ceci peut être utile avant de composer des images mono (les zones qui ne sont pas communes à toutes les images sont recadrées).

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

Alignement sur 1-2-3 étoiles

Lorsque les images contiennent peu d'étoiles, par exemple dans le cas des images de ciel profond rapide où le temps d'exposition est inférieur à une seconde. Il est possible que l'algorithme d'alignement global échoue, même si vous modifiez les paramètres de détection dans la fenêtre *PSF Dynamique*. Il peut alors être intéressant de faire une détection manuelle des étoiles que l'on veut aligner. C'est l'intérêt de l'algorithme d'alignement de 1, 2 ou 3 étoiles.

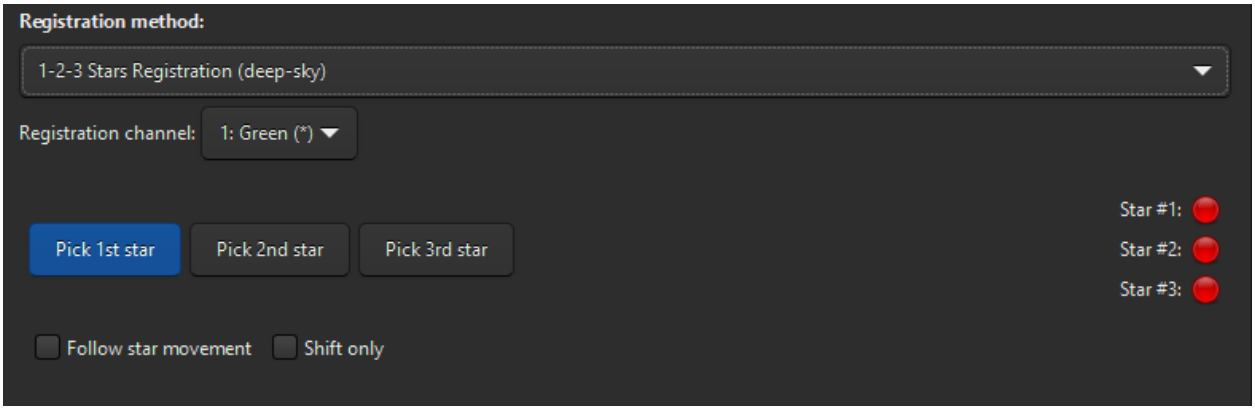


Fig. 25 – Option d'alignement sur 1-2-3 étoiles

Le principe de cette méthode est de dessiner une zone de sélection autour d'une étoile et de cliquer sur le bouton *Pointer l'étoile 1*, et ainsi de suite.

- Si une seul étoile est sélectionnée, seul la translation entre les images sera calculé. En conséquence, le bouton *Translation seulement* est automatiquement sélectionné. Les valeurs de translation sont alors stockées dans le fichier `seq`.
- Si deux ou trois étoiles sont sélectionnées, la rotation peut être calculée et appliquée pour créer une nouvelle séquence. Cependant, si l'option *Translation seulement* est sélectionnée, ce qui n'est pas obligatoire, seuls les décalages seront calculés.

L'option *Suivre le mouvement de l'étoile* utilise la position des étoiles trouvées dans l'image précédente comme nouveau centre pour l'alignement de l'image actuelle. Cela permet de réduire la zone de sélection, d'accélérer l'alignement et de tenir compte de la dérive ou des images comportant un grand nombre d'étoiles.

Avertissement

L'activation de cette option implique que l'alignement n'est pas parallélisé et qu'il s'exécute sur un seul cœur du processeur.

Alignement par motif de l'image (planétaire - disque entier)

Il s'agit d'un simple alignement par translation en utilisant la méthode de corrélation croisée dans le domaine spatial.

Cette méthode est rapide et est utilisée pour aligner les films *planétaires*, dans lesquels des informations contrastées peuvent être observées sur de grandes zones de l'image. Elle peut également être utilisée pour l'alignement d'images du ciel profond. Néanmoins, il faut garder à l'esprit qu'il s'agit d'une méthode d'alignement en un seul point, ce qui la rend peu adaptée à l'alignement de planètes en haute définition. Mais elle permet d'ancrer efficacement les images pour stabiliser la séquence. Il suffit de faire une sélection autour de l'objet (la planète par exemple) et de s'assurer que son mouvement pendant la séquence est contenu dans la sélection. Seule la translation peut être calculée avec cette méthode.

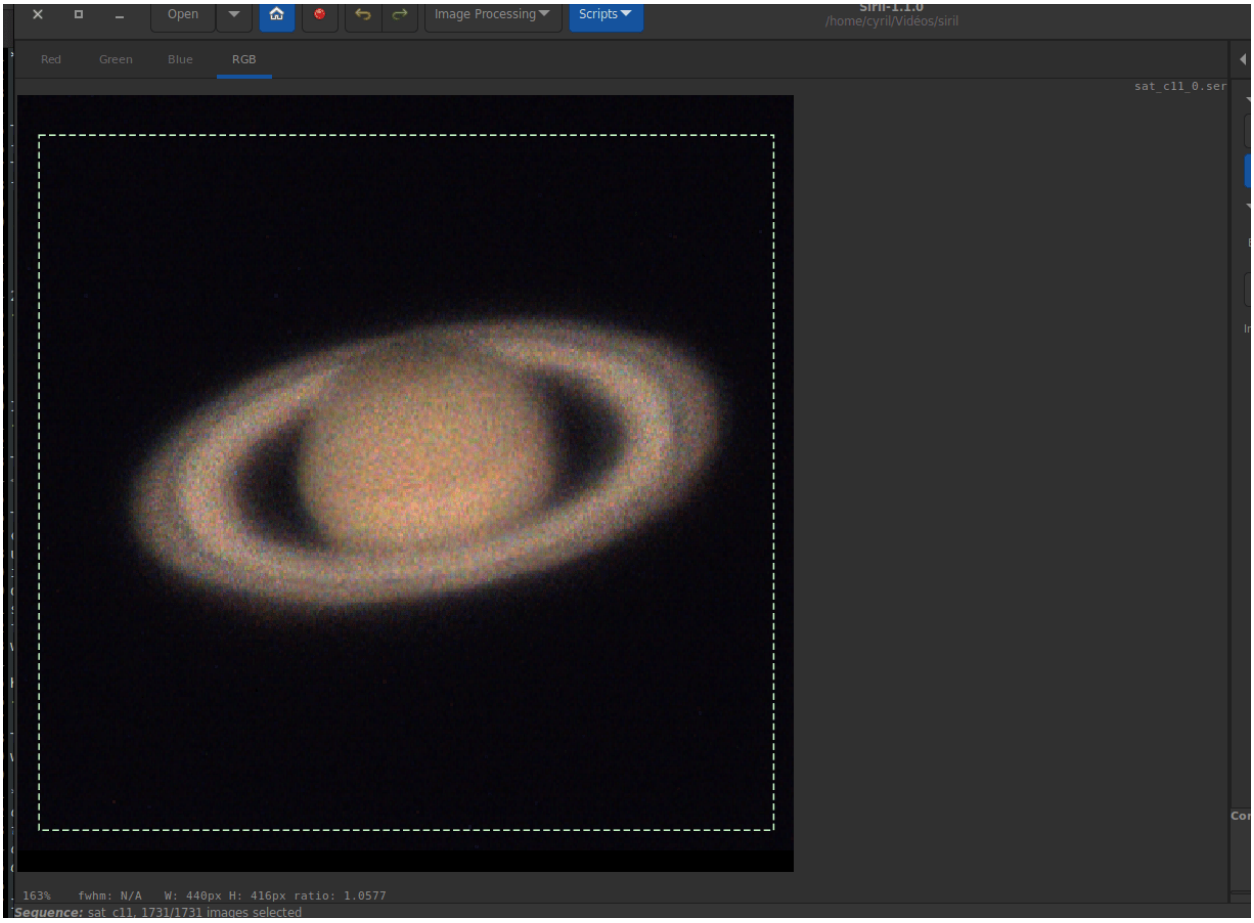


FIG. 26 – Option d'alignement par motif

KOMBAT

Cette méthode provient de la bibliothèque OpenCV, une bibliothèque largement utilisée dans Siril. Ils expliquent :

Il suffit de faire glisser l'image modèle sur l'image d'entrée (comme dans la convolution 2D) et de comparer le modèle et le patch de l'image d'entrée sous l'image modèle. Plusieurs méthodes de comparaison sont implémentées dans OpenCV. (Vous pouvez consulter la documentation pour plus de détails). Il renvoie une image en niveaux de gris, où chaque pixel indique dans quelle mesure le voisinage de ce pixel correspond au modèle.

En pratique, il suffit de tracer une sélection autour de l'objet (la planète par exemple) et de s'assurer que son mouvement au cours de la séquence est contenu dans la sélection. Seule la translation peut être calculée avec cette méthode.

Alignement Comète/Astéroïdes

L'outil d'alignement cométaire fonctionne d'une façon très simple, en deux étapes.

1. Dans le sélecteur d'image, sélectionnez la première image de la séquence, entourer le noyau cométaire, ensuite cliquer sur le bouton *Pointer objet dans #1*.
2. Ensuite sélectionnez la dernière image de la séquence, entourez le noyau cométaire, ensuite cliquez sur le bouton *Pointer objet dans #2*.

La vitesse de la comète Δx et Δy est calculé en pixel par heure si tout est ok.

Avertissement

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, via a 2-pass or astrometric registration.

Note

Pour fonctionner pleinement, les images doivent être horodatées.

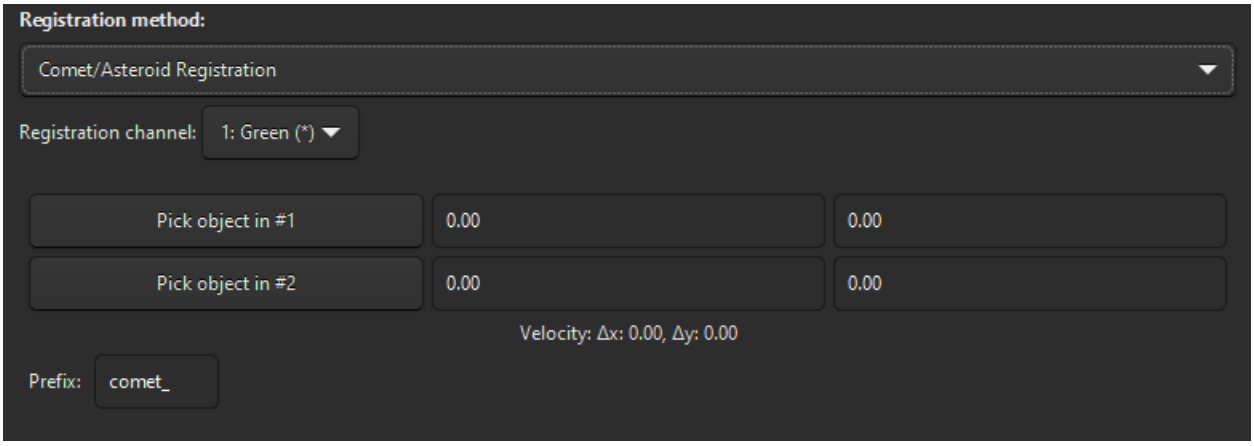


FIG. 27 – Options d'alignement pour une comète

Astuce

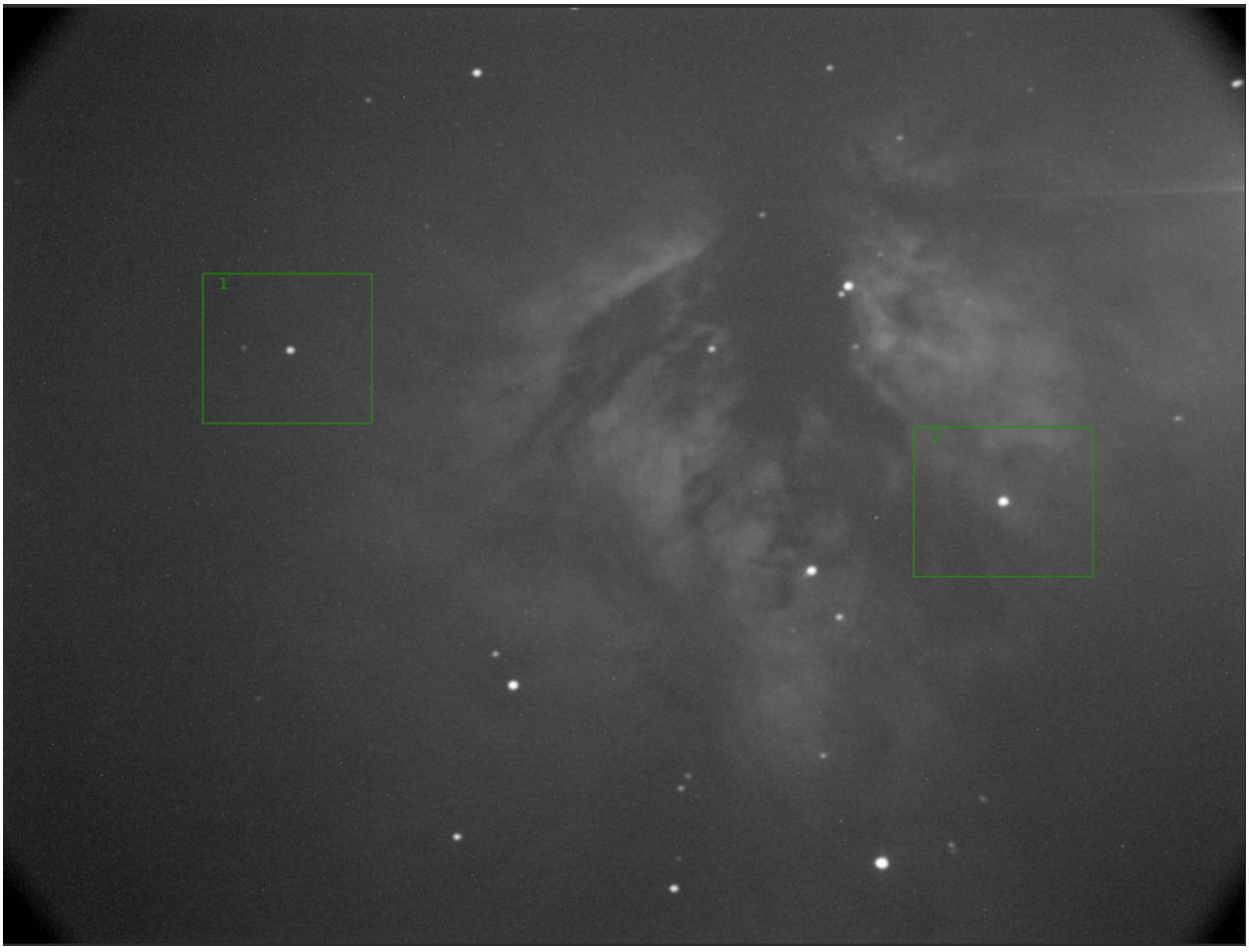
If the PSF detection fails to detect an object, it will return the center of the box that was drawn. This can be handy if you want to align on an object that is not visible on your subs. Use the *solar system object* annotations to plot the position of an asteroid and draw a box around the marker to pick its position.

This method will ouput a new sequence file, with the sequence name prepended with the *prefix* defined (`comet_` by default). This does not however create the new images but symlinks to the original images. For Windows users, please make sure you have enabled developer mode, otherwise, it will make hard copies. See also the note below *this table*.

Alignement manuel

Cette dernière méthode d'alignement est très particulière, ce qui explique sa position à part, et permet d'aligner les images manuellement. Bien entendu, seule la translation entre les images est autorisée.

La première chose à faire est de définir deux aperçus dans l'image. En cliquant sur le bouton *Définir 1er aperçu* (Définir la première prévisualisation), vous initialiserez la première prévisualisation. Il faut alors cliquer sur une zone de l'image, idéalement une étoile à proximité d'un bord de l'image pour définir la zone de prévisualisation. Un clic sur le second bouton *Définir 2d aperçu* permet de faire de même sur un second point.



Il est très important d'avoir une image de référence déjà définie avec le *Sélecteur d'image*. Par défaut, il s'agit de la première image. L'utilisateur est libre de choisir celle qu'il veut. Elle servira d'image de référence, vu par transparence, pour aligner les images manuellement avec les boutons numériques. Ensuite, parcourez les images une à une pour appliquer la même méthode à l'ensemble de la séquence.

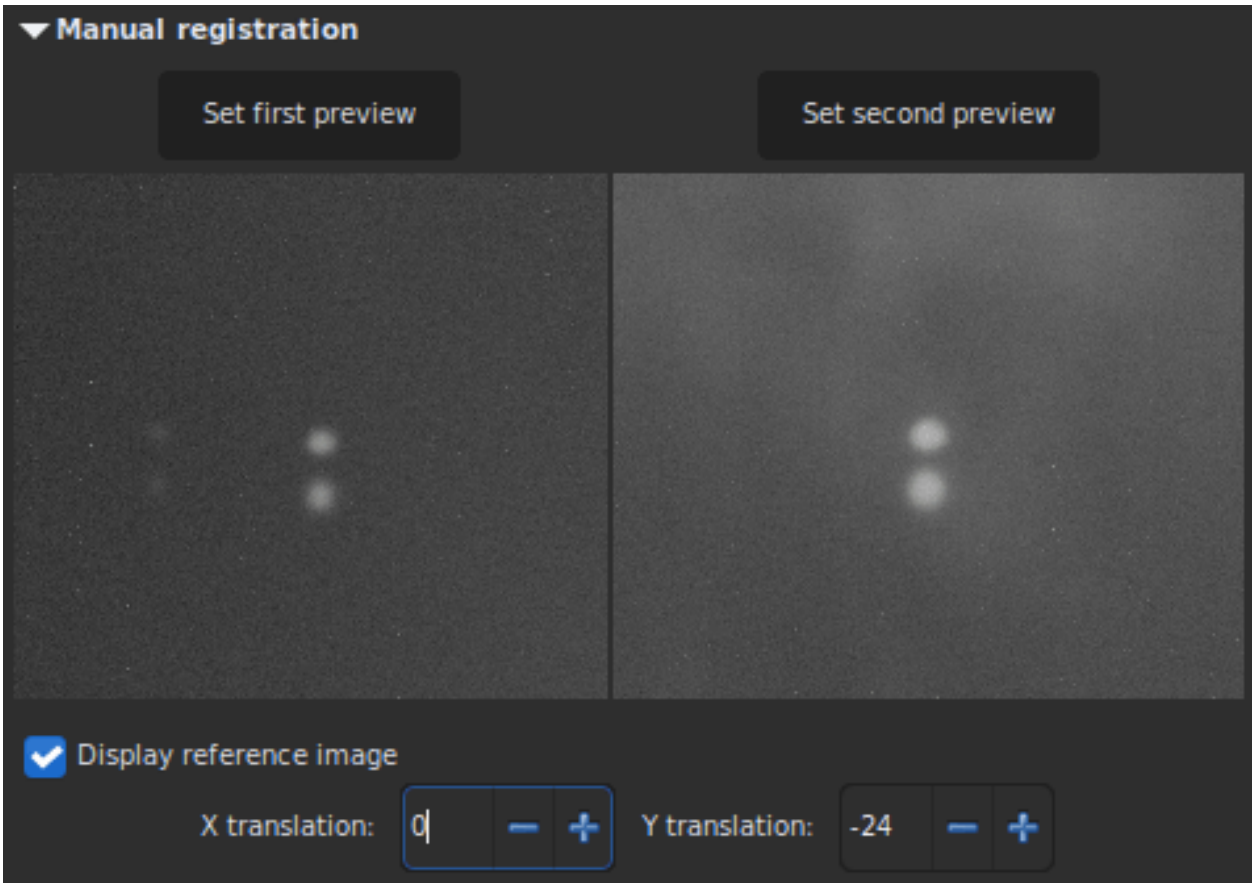


FIG. 28 – Le décalage en Y est trop important, les mêmes étoiles sur des images différentes ne se chevauchent pas.

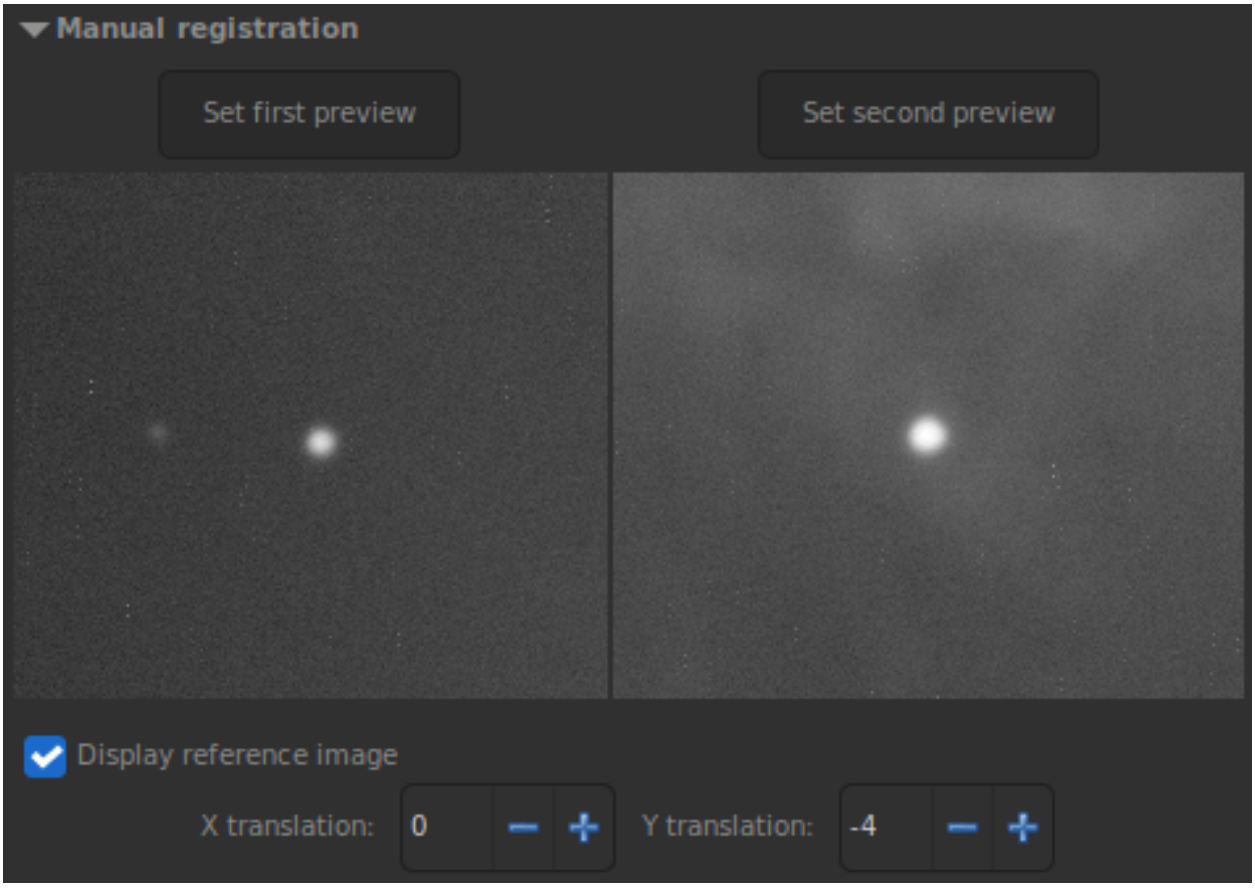


FIG. 29 – Les décalages X et Y semblent corrects. L'image actuelle est alignée sur l'image de référence.

Alignement astrométrique

Introduced in version 1.3, this is the prefered mode for assembling mosaics or images with little overlap. It can also be useful to register stacks issued from different set-ups (different optics, different cameras, different field of views etc...).

It does not have an entry in the registration method drop-down as the information to export the registered images has already been computed when *platesolving the sequence*. All you need to do is to *Apply Existing Registation*.

Undistortion will be applied as defined when platesolving the sequence, meaning if the images were plate-solved using a SIP order larger than 1, then undistortion will automatically be included. Unless you have a perfectly optically flat field, it is usualy a good idea to platesolve using SIP, as shsown below, with and without distortion correction.

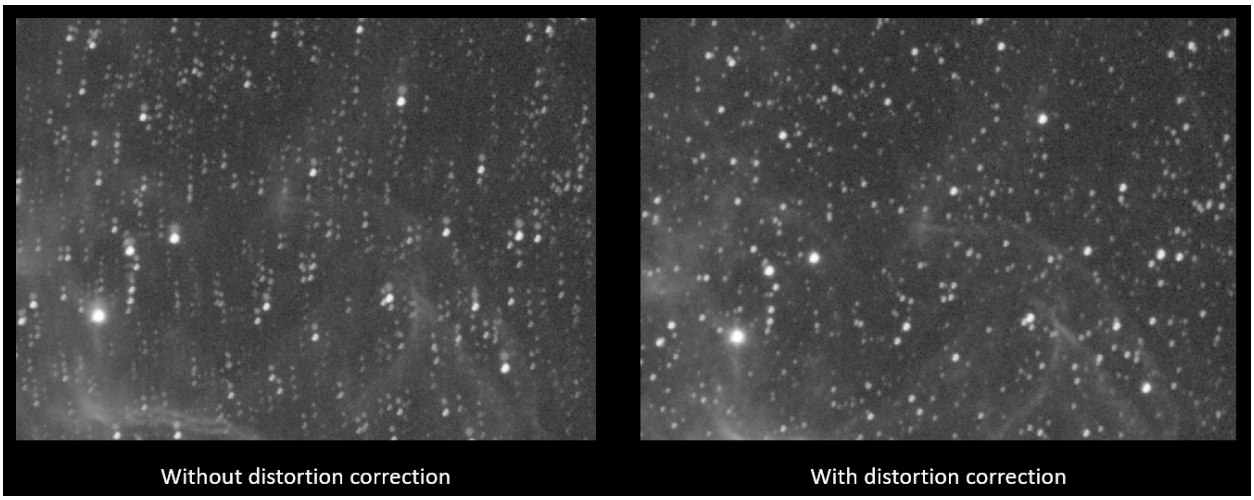
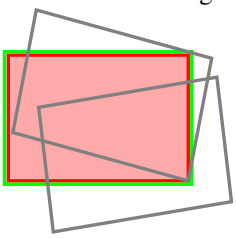


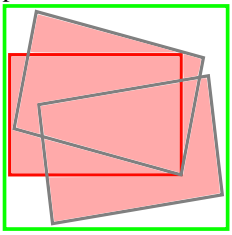
FIG. 30 – Effet de la suppression de la distorsion sur deux panneaux superposés après alignement

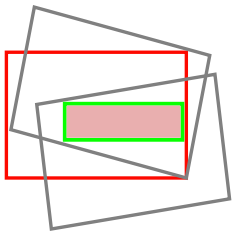
Appliquer Alignement Existant

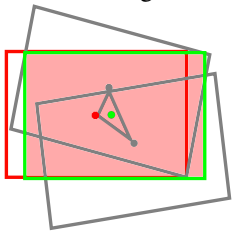
Il ne s'agit pas d'un algorithme, mais plutôt d'une commodité pour appliquer les données d'alignement précédemment calculées et stockées dans le fichier de séquence. La méthode d'interpolation peut être sélectionnée dans la section*Sortie d'alignement*. Vous pouvez également utiliser le filtrage d'image pour éviter d'aligner des images inutiles, comme dans l'empilement*Rejet des images*. Il y a aussi une option*Drizzle* pour appliquer l'alignement en utilisant le drizzle au lieu de l'interpolation. Voir la section*Drizzle* pour plus de détails.

Quatre méthodes de cadrage sont disponibles :

—  : **Actuelle** utilise l'image actuelle comme image de référence. Il s'agit du comportement par défaut.

—  : **maximum** (zone de délimitation) ajoute un bord noir autour de chaque image si besoins affin qu'aucune partie de l'image soit recadré lors de l'alignement.

—  : **minimum** (zone commune) recadre chaque image sur la zone qu'elle a en commun avec toute les images de la séquence.

—  : **centre de gravité** détermine la meilleur position de cadrage en tant que centrer de gravité (cog) de toutes les images.

> **Astuce**
>
> Introduced in Siril 1.3, `max` mode does not export images with black borders, that encompass the full resulting image. The images are exported with the necessary projection and the relative shifts required to compose the final image is kept in the the resulting seq file.

Estimate button will launch the framing computation, without actually exporting the images. This information can be interesting to know in advance the size of the exported images. This accounts for the **framing** method selected and the **scaling** factor chosen in the *Output Registration*.

En appuyant sur*Estimer*, la console affichera une sortie comme celle-ci :

```
Output image: 7893 x 5254 pixels (assuming a scaling factor of 1.30)
```

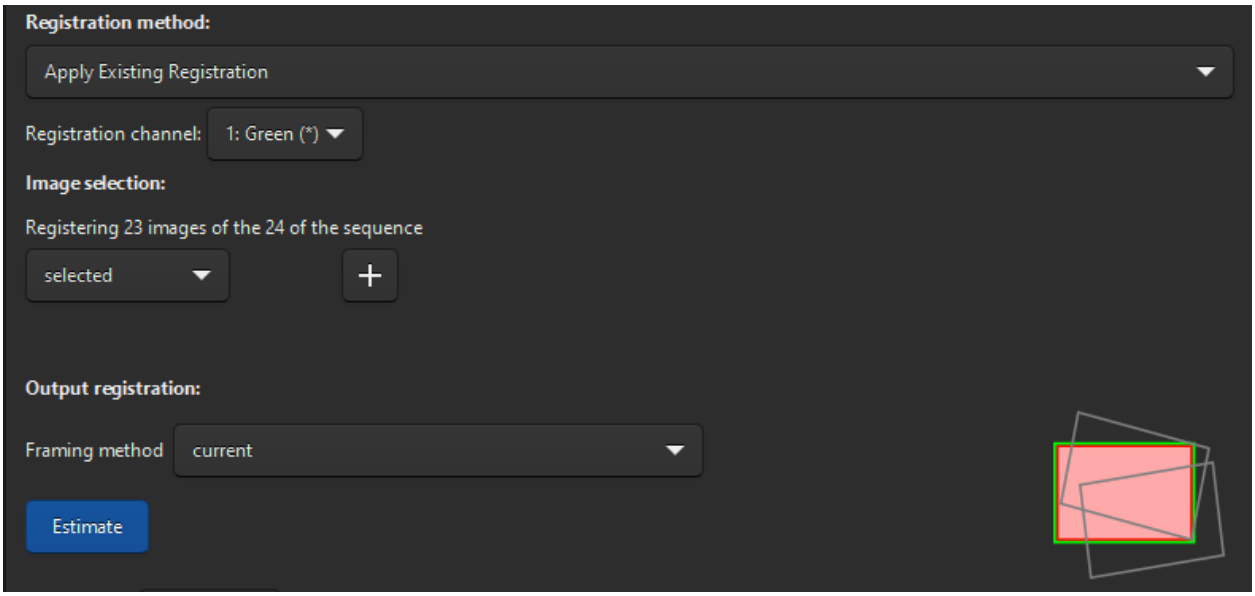


Fig. 31 – Option d'Appliquer Alignement Existant

Ligne de commande Siril

```
seqapplyreg sequencename [-prefix=] [-scale=] [-layer=] [-framing=]
seqapplyreg sequencename ... [-interp=] [-noclamp]
seqapplyreg sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
seqapplyreg sequencename ... [-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-
↪filter-round=value[%|k]] [-filter-bkg=value[%|k]] [-filter-nbstars=value[%|k]] [-
↪filter-quality=value[%|k]] [-filter-incl[uded]]
```

Applique des transformations géométriques aux images de la séquence donnée en argument afin qu'elles puissent être superposées à l'image de référence, en utilisant les données d'alignement précédemment calculées (voir REGISTER).

Le nom de la séquence de sortie commence par le préfixe **"r_"** sauf indication contraire spécifié par l'option **-prefix=**.

L'alignement est réalisé sur la première couche pour laquelle les données existe pour une image RVB sauf si spécifié par l'option **-layer=** (0, 1 ou 2 pour R, V et B respectivement).

The output images can be rescaled by passing a **-scale=** argument with a float value between 0.1 and 3.

Le cadrage automatique de la séquence de sortie peut être spécifié en utilisant le mot-clé **-framing=** suivi de l'une des méthodes de la liste { current | min | max | cog } :

-framing=max (bounding box) will project each image and compute its shift wrt. reference image. The resulting sequence can then be stacked using option **-maximize** of STACK command which will create the full image encompassing all images of the sequence.

-framing=min (zone commune) recadre chaque image dans la zone qu'elle a en commun avec toutes les images de la séquence.

-framing=cog détermine la meilleur position de cadrage comme centre de gravité (cog) de toutes les images.

Option d'interpolation de l'image :

Par défaut, les transformations sont appliquées pour aligner les images à l'aide de l'interpolation.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.

La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**.

Options de drizzle pour les images :

Sinon, les images peuvent être exportées à l'aide de l'algorithme HST drizzle en passant l'argument **-drizzle** qui peut prendre les options supplémentaires suivantes :

-pixfrac= défini la fraction de pixel (par défaut =1.0).

L'argument **-kernel=** défini le noyau de drizzle et doit être suivi par l'un des arguments suivant : **point**, **turbo**, **square**, **gaussian**, **lanczos2** or **lanczos3**. La valeur par défaut est **square**.

L'argument **-flat=** spécifie un flat maître pour pondérer les pixels d'entrée drizzlés (la valeur par défaut est sans flat).

Filtrage des images :

Les images qui vont être aligner peuvent être sélectionnées en utilisant des filtres, tel que celles sélectionné ou avec la meilleur FWHM, avec l'une des option **-filter-***.

Links : *register*, *stack*

Avec le filtrage étant l'un de ceux-ci, sans ordre ni nombre particulier :

```
[-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]]
```

Les meilleurs images de la séquence peuvent être empilées en utilisant les arguments de filtrage. Chacun de ces argument peu supprimer les mauvaises images sur la base des propriétés que leur nom contient, récupéré depuis les données d'alignement, avec l'un des trois types de valeur d'argument :
- 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.

Il est également possible d'utiliser des images sélectionnées manuellement, soit précédemment à partir de l'interface graphique utilisateur (GUI), soit avec les commandes select ou unselect, en utilisant l'argument **-filter-included**.

9.3.3 Sortie d'alignement

This frame contains all the output elements for the sequence. You can choose between Interpolation and Drizzle to export the images.

Avertissement

They may not be both available depending on the nature of the input sequence :
- pour les séquences mono, interpolation et Drizzle peuvent être sélectionné
- pour les séquences CFA (image couleur non débayerisé), seul Drizzle est disponible
- pour les séquences RGB (images couleurs débayerisé), seul l'interpolation est disponible

Les deux méthodes partagent les mêmes options :
- *Scaling*, a value between 0.1 and 2, that is used to rescale the output images.
- *Prefix*, the prefix that will be preprended to form the exported sequence name (defaults to `r_`).

Interpolation

Tthe 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 :
- Plus Proches Voisins
- Bilinéaire
- Bicubique
- Zone de relation des pixels
- Lanczos-4
- Aucun

The best efficient interpolation methods are generally bicubic and Lanczos-4 (used by default). However, they usually require the *Clamping interpolation* option to be enabled to avoid ring artifacts around the stars, which makes the inter-

polation operation twice as slow. We still enable it by default because it gives a significant improvement in resolution, with often a 8% smaller in FWHM compared to Siril 1.0 default method of pixel area. It is possible the clamping is not useful for your images, we recommend you to compare the results with your images.

Le cas particulier de **Aucun** est réservé au cas de l'alignement global et de l'application de l'alignement existant. Si vous souhaitez exporter ou sauvegarder une séquence qui ne contient que de la translation, sans utiliser d'interpolation (afin de ne pas modifier les valeurs des pixels), vous devez sélectionner **Aucun**.

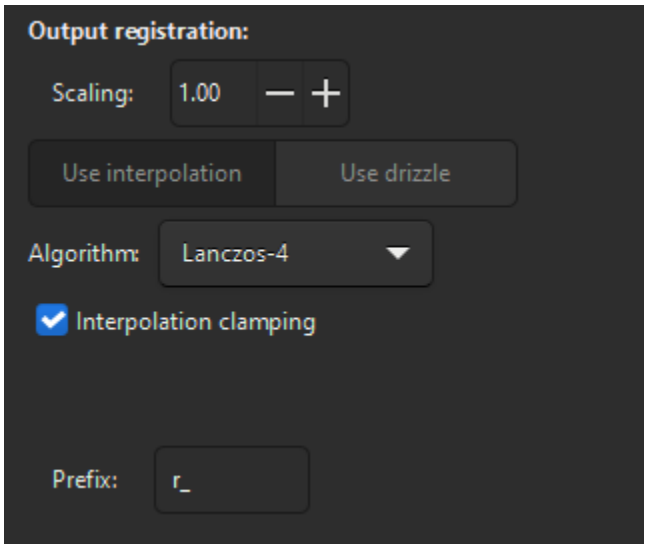


FIG. 32 – Options de sortie pour l'interpolation

Drizzle

Le bouton*Drizzle* active l'algorithme drizzle pour le traitement de cette séquence. Voir la section*Drizzle* pour plus de détails.

Avertissement

La contrepartie de cette technique est que la quantité de mémoire et d'espace disque nécessaire à la création et au traitement des images drizzle est multipliée par le carré du facteur d'échelle du drizzle.

Solution astrométrique des images alignées

When registered images are exported, they inherit the astromertic soluion of the reference image if any. Otherwise, their previous solution is wiped. Obviously, the new solution for each image accounts for the transformations it has undergone through the registration process.

In case a distortion solution is found in the reference image but no undistortion was applied when computing the transformations - for instance if you have not selected any `Distortion` from the drop-down in global or 2-pass methods, a warning will be displayed in the Console. The distortion information will be kept in the registered images. In case they are significant, you may see their effect when stacking. In which case, you will need to register again using a distortion specification.

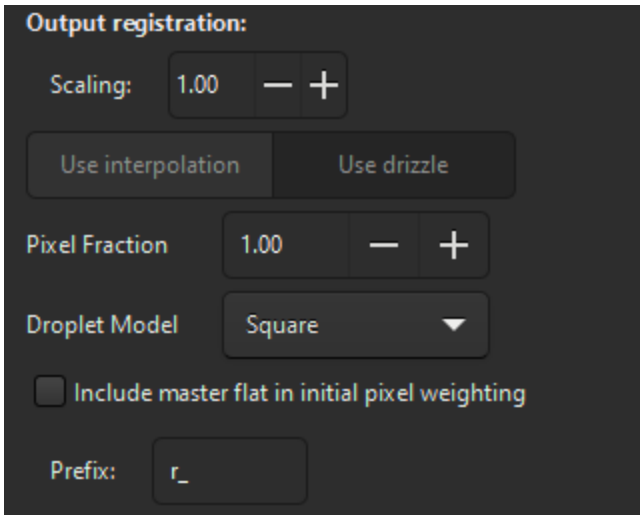


FIG. 33 – Options de sortie pour drizzle

9.3.4 Références

9.4 Drizzle

La reconstruction linéaire à pixels variables, plus communément appelée Drizzle, a été développée pour le télescope spatial Hubble (HST) par Andy Fruchter et Richard Hook [FruchterHook1997], initialement dans le but de combiner des images avec dithering du Hubble Deep Field North (HDF-N). Cet algorithme peut être considéré comme un ensemble continu de fonctions linéaires qui varient doucement entre la technique de combinaison linéaire optimale (entrelacement) et la technique standard. Cela permet souvent d'améliorer la résolution et de réduire le bruit corrélé, par rapport aux images produites, en utilisant uniquement la technique standard.

Il existe une excellente page qui résume la technique et fournit une bonne représentation graphique de la manière dont les données des pixels "descendent" de la grille de pixels d'entrée grossière vers une grille de pixels de sortie plus fine ici.

La mesure dans laquelle l'algorithme s'écarte de l'entrelacement et se rapproche de l'empilement standard dépend de la qualité du sous-échantillonnage de la PSF par les décalages des images d'entrée. En pratique, le comportement de l'algorithme Drizzle est contrôlé par l'utilisation d'un paramètre appelé fraction de pixel, qui représente la quantité de pixels d'entrée qui sont réduits avant d'être mappés sur le plan de l'image de sortie. À une fraction de pixel de 0, l'algorithme Drizzle est équivalent à un entrelacement pur ; à une fraction de pixel de 1, il est équivalent à un empilement standard.

Pour comprendre la différence entre les méthodes d'alignement par drizzle et par interpolation, examinons tout d'abord le fonctionnement de la méthode d'interpolation standard. Les données d'alignement prennent la forme d'une matrice d'homographie 3x3, qui code une transformation linéaire à 8 degrés de liberté d'un ensemble de coordonnées à un autre (c'est-à-dire de chaque image à l'image de référence). Cette transformation est utilisée pour faire correspondre les valeurs de chaque pixel de chaque image d'entrée à l'endroit correct de l'image de sortie, en alignant cette dernière sur l'image de référence. L'alignement réel utilise une méthode d'interpolation, qui peut être sélectionnée dans les options d'alignement. L'interpolation entraîne un étalement de la fonction d'étalement du point, en particulier lors de la mise à l'échelle des images. Elle peut également donner lieu à des artefacts, bien que Siril mette en œuvre un mécanisme pour minimiser ce phénomène.

Le drizzle, en revanche, transforme chaque pixel de l'image d'entrée en une "gouttelette" et la projette à travers une grille sur l'image de référence de sortie. Chaque gouttelette a une taille, et en choisissant une grille de pixels de sortie plus grande mais une taille de gouttelettes plus petite, vous pouvez obtenir une meilleure résolution **si** votre jeu d'images est sous-échantillonné. (Si votre échantillonnage est correct pour le pouvoir de résolution de votre télescope, le drizzle

ne peut pas produire de détails au-delà de la limite de diffraction). Cela se fait au prix d'une augmentation du bruit de l'image : comme chaque gouttelette "peint" une zone plus petite dans l'image de sortie, la couverture moyenne des gouttelettes par pixel de sortie dans la pile finale est réduite.

Notez que le drizzle ne remplace pas l'ensemble du processus d'alignement : vous pouvez toujours utiliser l'alignement global des étoiles, l'alignement 1-2-3 des étoiles, l'alignement des comètes ou toute autre méthode d'alignement de votre choix avant le drizzle : il s'agit uniquement d'une alternative à la méthode d'interpolation utilisée lors de l'application de l'alignement.

Avertissement

The Drizzle process involves taking multiple frames and drizzling them into an output frame. The input is the set of frames and their WCS solutions, and the output is the single drizzled image. In Siril the process is split into Drizzle registration and subsequent stacking, however the intermediate artefact (the set of images representing the individual images drizzled onto the common output grid but not yet stacked) is not something that has significance in its own right. Individual frames in the drizzled sequence can **and will** look weird : the reference frame often looks different to the others, especially when drizzling CFA data, as a result of its special property of having zero shift with respect to the output frame, and other frames may show Moiré patterns. **Don't panic !** This is simply an intermediate stage in the overall Drizzle process, and all these apparent oddities will be resolved in the stacking stage.

9.4.1 Utilisation du drizzle

Il y a trois raisons principales pour lesquelles vous pouvez préférer le drizzle à une méthode d'interpolation pour appliquer l'alignement.

- **Amélioration de la résolution** Si votre image est significativement sous-échantillonnée, vous pouvez obtenir une amélioration de la résolution avec le drizzle que vous ne pourriez pas obtenir avec l'option d'alignement*Interpolation, mise à l'échelle x2*.
- **Imagerie couleur** Si vos images présentent un motif CFA (c'est-à-dire si vous utilisez une caméra couleur ou un reflex numérique), le drizzle apporte une amélioration significative par rapport au dématriçage par interpolation. Ce procédé est parfois appelé *Bayer Drizzle*, mais il s'agit en fait exactement du même processus. Lors du drizzle d'une image CFA, la couleur CFA de la gouttelette détermine le canal de l'image de sortie sur lequel elle atterrit, alors que lors du drizzle d'une image mono, toutes les gouttelettes atterrissent dans le (seul) canal de sortie mono. Le drizzle d'images CFA évite les artefacts qui se produisent avec tous les algorithmes de dématriçage, ce qui améliore les caractéristiques du bruit lorsque les images sont fortement étirées. Cela permet d'améliorer la réduction du bruit et la déconvolution pour les séquences CFA drizzle par rapport aux images CFA dématricées et alignées, et améliore l'aspect du fond de ciel.
- **Éviter les artefacts** Il est possible d'effectuer un drizzle sur une séquence en utilisant l'échelle = 1, la fraction de pixel = 1 et d'obtenir essentiellement le même résultat qu'en appliquant l'alignement avec l'une des méthodes interpolées. Vous pouvez envisager cette solution si vous constatez des artefacts d'interpolation avec la méthode d'interpolation standard (bien que ces artefacts soient généralement supprimés de manière efficace par la fonction de clamping). Notez que le drizzle peut produire des artefacts différents (voir le titre "Quelques problèmes communs" ci-dessous), mais ceux-ci peuvent être complètement évités en choisissant le noyau de drizzle ou en ayant un plus grand nombre d'images d'entrée, et sont habituellement parfaitement gérés par l'empilement.

9.4.2 Limitations du drizzle

- Le drizzle est un peu plus lent que l'interpolation, en particulier en utilisant le noyau carré qui est le noyau par défaut. Si vous utilisez du matériel plus ancien ou plus lent, vous préférerez peut-être l'ancienne méthode.
- Lorsqu'elle est utilisée pour le suréchantillonnage, la méthode Drizzle permet d'améliorer la résolution au prix d'une augmentation du bruit de l'image. Par conséquent, vous pouvez souhaiter collecter plus de temps d'intégration lorsque vous utilisez la technique du Drizzle que lorsque vous utilisez une méthode de suréchantillonnage basée sur l'interpolation.

— Le problème ci-dessus est particulièrement vrai pour les images CFA. Considérons que seulement 1 pixel sur 2 est vert, et que seulement 1 pixel sur 4 est rouge ou bleu. Par conséquent, pour les canaux rouges ou bleus, le drizzle CFA implique déjà le même niveau de réduction de la couverture des gouttelettes qu'un drizzle à échelle 2 fois supérieure. Si vous augmentez l'échelle en plus, vous avez besoin d'autant de gouttelettes que pour un drizzle à x4 ! C'est pourquoi il est généralement recommandé de faire un drizzle des images CFA à l'échelle = 1.

9.4.3 Comparaison

L'image suivante montre une comparaison entre le drizzle et l'ancienne méthode de suréchantillonnage. L'image est une image Ha extraite d'une session dapteur couleur avec un filtre double bande. À gauche, vous pouvez voir le résultat du script OSC_Extract_HaOIII, qui extrait les données Ha capturées par les pixels rouges dans la matrice de Bayer du capteur couleur en tant qu'image de demi-taille et utilise la mise à l'échelle d'OpenCV avec l'interpolation lanczos4 pour produire une image qui correspond à la taille de l'image OIII.

On the right you can see the result of the updated script (available through the siril-scripts repository), which extracts the Ha data captured by the red pixels in the OSC Bayer matrix as a half-size image and drizzles it using scale = 2.0, pixel fraction = 0.5, to produce an image that matches the size of the OIII image.

En l'observant à l'échelle de 100 %, il est clair que le stack drizzle restaure une grande partie de la résolution du système optique qui est sous-échantillonnée par les pixels rouges espacés dans la matrice de Bayer : elle semble beaucoup plus nette, et les chiffres le confirment : la fwhm moyenne dans l'image de gauche est de 3,59, tandis que dans l'image de droite, elle est de 3,25.

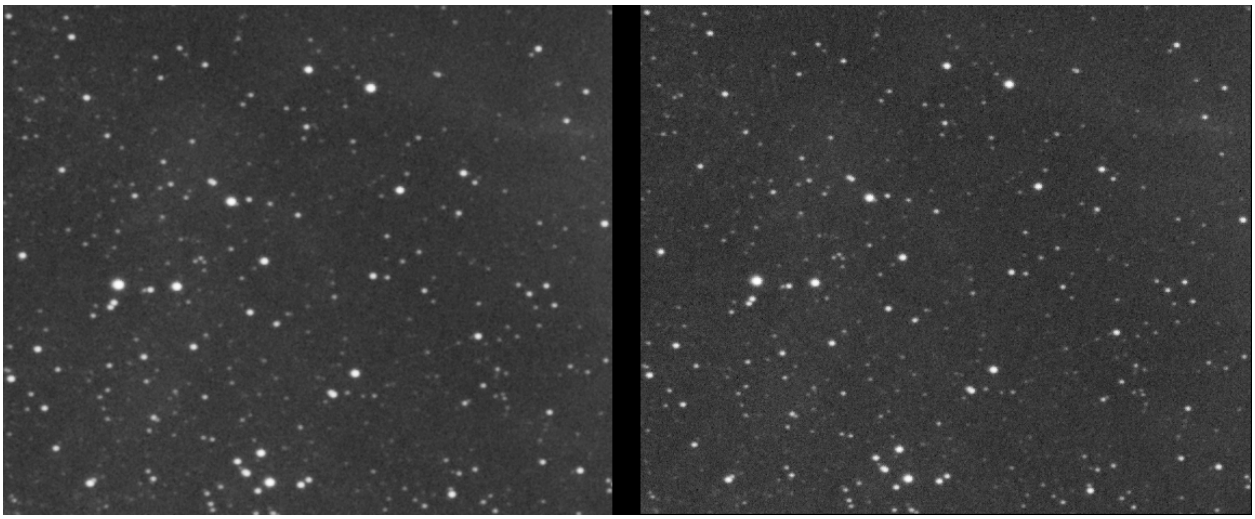


FIG. 34 – Comparaison entre l'interpolation, le suréchantillonnage et le drizzle

9.4.4 Flux de travail et interface utilisateur

9.4.5 Flux de travail mono

Calibration

Pour les images mono, rien ne change dans l'onglet de calibration. Calibrez comme vous le feriez normalement.

Alignement

Drizzle

Échelle

L'échelle définit l'échelle de l'image de sortie du drizzle par rapport à l'image d'entrée. Une échelle typique pour une image mono sous-échantillonnée est de 2,0. Cela signifie que l'image d'entrée sera appliquée sur une grille de pixels de sortie dont la résolution est deux fois plus élevée. (Si votre image de référence en entrée est de 1024 x 512 pixels, votre image de sortie sera de 2048 x 1024 pixels). Remarque : étant donné que l'image représente la même zone du ciel, bien qu'il y ait **deux fois** plus de pixels le long de chaque axe dans l'image de sortie, chaque pixel de sortie est en fait **moitié moins large** et **moitié moins haut**.

> **Astuce**
>
> Plus l'échelle est grande, plus l'image de sortie est clairsemée et moins il y a de pixels empilés dans chaque pixel de sortie. Il en résulte une image plus bruitée : **le gain de résolution apporté par le drizzle se fait au détriment du bruit**. Ce phénomène doit être atténué par l'utilisation d'un temps d'intégration global plus important que celui dont vous auriez besoin sans l'application de la méthode drizzle à une plus grande résolution.

Fraction de pixel

Pixel Fraction sets the size of the droplet taken from the input grid. Consider a drizzle scale of 2.0 : since the output pixels are half as wide and half as tall, that means that in order for each input pixel "droplet" to be the same size as an output pixel it should be shrunk to half the linear dimensions. This is a pixel fraction of 0.5. A good rule of thumb is that the pixel fraction should be roughly the reciprocal of the drizzle scale (with some kernels it helps to set it a little bigger than this, in order to reduce pixels that receive zero input from any drizzled droplets).

Il est possible d'expérimenter avec la fraction de pixel : une fraction de pixel plus importante signifie que chaque gouttelette d'entrée influencera plus de pixels de sortie. À l'inverse, une fraction de pixels plus petite signifie que chaque gouttelette d'entrée influencera moins de pixels de sortie. Le noyau "point" est un cas particulier où la fraction de pixel est nulle (et avec ce noyau sélectionné, le réglage de la fraction de pixel n'a pas d'effet).

Modèle de la gouttelette

L'implémentation du drizzle de Siril fournit plusieurs modèles de gouttelettes :

— **Square**. This models the droplet as a square droplet aligned exactly with the input pixel. It is accurately mapped to the output reference frame, and it works well at any scale and pixel fraction. This is the default kernel.

> **Astuce**
>
> The Square kernel is mathematically flux preserving by construction : each input pixel's flux is distributed into output pixels in exact proportion to area overlap. Total counts are preserved identically. This makes it both robust and ideal for applications where you need strict surface brightness conservation (e.g. diffuse structures or extended sources).

— **Point**. This models the droplet as a point at the center of the input pixel. It is the limit of the square kernel with a very small pixel fraction. This will prodce the lowest correlated noise but is prone to leaving holes unless you have very large numbers of well-dithered frames. If you find holes or artefacts in your final stack, use the Square kernel instead. This kernel is very fast.

— **Turbo**. This is a simplification of the Square kernel. It assumes that rotation between the input and output reference is negligible. This results in a much faster computation, but is approximate. It is a "quick and dirty" kernel originally intended for generating intermediate products within the HST workflow, but it may be of some use in lucky imaging due to its speed relative to Square.

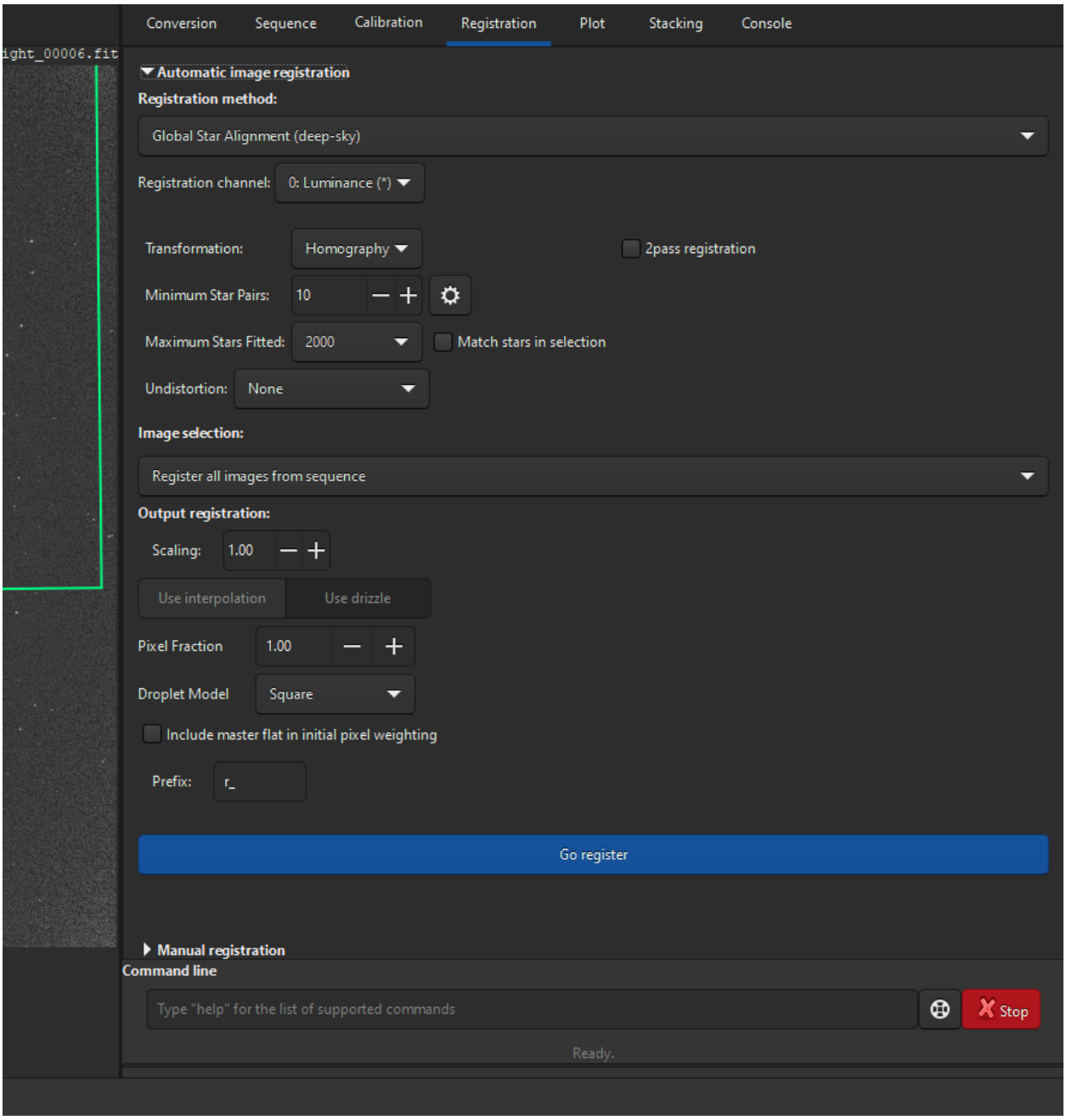


FIG. 35 – Onglet d'alignement montrant les paramètres du drizzle

— **Gaussian**. This models the droplet as a Gaussian centered on the center of the input pixel with a FWHM equal to *pixfrac*. The droplets in the Gaussian model are round and are recommended where improved preservation of PSFs (star shapes) is important.

> **Astuce**
>
> [Avila2015] recommends the Gaussian kernel for applications involving point source photometry. Despite not being mathematically flux preserving, the Gaussian kernel reduces the blockiness typical of square kernels (and avoids the high frequency ringing characteristic of Lanczos kernels, producing smooth, centrally peaked PSFs that are closer to analytical forms used in photometry. While Gaussians don't preserve flux perfectly the trade-off is much better behaved PSFs, which is more important for extracting accurate stellar photometry. It may also be more aesthetically pleasing in a final image.

— **Lanczos2** and **Lanczos3**. These kernels model the droplet as a Lanczos function centered on the center of the input pixel.

> **Avertissement**
>
> The Lanczos kernels are specially designed for resampling an image from one World Coordinate System (WCS) onto another **at the same scale**. They should only be used with scale == pixfrac == 1.0.

Lanczos kernels provide a good option for single-image registration at native scale : Lanczos3 produces tigher average FWHM than any other kernel or OpenCV interpolation, but it is significantly slower than other kernels.

Pondération initiale des pixels

When a droplet lands on the output pixel grid, it may cover more than one output pixel. In fact, one output pixel may be covered by multiple droplets, by only a fraction of a droplet or even by no droplets at all. The contribution of each input pixel can be weighted by the master flat, so that pixels from areas with higher SNR (less vignetting) are weighted more highly. Unless you have peculiar flats this generally makes no practical difference.

Pour activer le master flat, cochez la case*Inclure le master flat dans la pondération initiale des pixels*.

> **Avertissement**
>
> Le master flat doit être défini dans l'onglet*Calibration* !

C'est parti pour le drizzle !

Once all the options are set, click the *Go register* button.

Empilement

Vous pouvez maintenant empiler votre séquence en drizzle comme d'habitude. Notez que pour certaines combinaisons d'échelle de drizzle et de taille de gouttelettes, certains modèles de rejet fonctionneront mieux que d'autres. En particulier, si vous avez un nombre important de pixels "zéro" ou nuls, il y aura moins de valeurs à utiliser pour le rejet. MAD peut être une bonne solution à essayer si votre méthode de rejet habituelle ne fonctionne pas.

Le GIF ci-dessous montre une comparaison d'un stack de 37 images, dans un cas avec alignement appliqué par interpolation et dans l'autre cas avec alignement appliqué à l'aide de drizzle. Il est clair que le stack réalisé avec les données "drizzle" est nettement plus nette que celles utilisant les données alignées par interpolation.

Fig. 36 – Comparaison de l'alignement appliqué avec le drizzle et avec l'interpolation. Cliquez sur l'image pour l'agrandir.

9.4.6 Flux de travail CFA

Calibration CFA

Pour les images couleur, décochez la case*Dématricer avant sauvegarde*. Cela représente un changement par rapport aux flux de travail précédents, mais pour le drizzle, il est essentiel que le motif CFA soit préservé dans la séquence d'entrée.

Alignement CFA avec drizzling

You can use drizzle directly after a *Global Registration* or by *applying existing registration* for all the other registration methods that do not export images directly.

Échelle CFA

L'échelle définit l'échelle de l'image de sortie du drizzle par rapport à l'image d'entrée. Dans les images des caméras couleur, chaque pixel n'enregistre qu'une seule couleur : rouge, vert ou bleu. Une matrice de filtres de couleur (CFA) est appliquée aux pixels et détermine quels pixels répondent aux longueurs d'onde rouges, vertes et bleues. Ainsi, tous les pixels sont répartis de manière éparse par rapport à un capteur mono, dans lequel tous les pixels sont sensibles à la lumière qui passe à travers le filtre. Dans les cas Bayer CFA et CFA X-Trans, les pixels rouges et bleus sont particulièrement peu nombreux dans les images d'entrée. Par conséquent, si l'on augmente l'échelle du drizzle au-delà de 1,0, il faudra encore plus d'images pour obtenir une couverture suffisante du drizzle et atteindre un niveau de bruit acceptable.

Pour une application typique de capteur couleur où le seeing correspond bien à l'échantillonnage nominal du capteur, il est recommandé d'appliquer le drizzle CFA avec échelle = 1.0 et fraction de pixel = 1.0. Cela rétablira la résolution dans chaque canal de couleur (qui est effectivement sous-échantillonné en raison de l'espacement des pixels colorés) et évitera les artefacts de dématriçage conventionnels. Si vous souhaitez également suréchantillonner l'image en utilisant l'échelle > 1.0, sachez que les pixels disponibles dans chaque canal deviendront encore plus rares et que vous aurez besoin d'encore plus de données pour assurer une couverture adéquate et contenir le bruit à un niveau acceptable.

> **Astuce**
>
> Pour le drizzle CFA, commencez par l'échelle = fraction de pixel = 1,0.

Fraction de pixel CFA

La fraction de pixel définit la taille de la gouttelette prélevée sur la grille d'entrée. Les mêmes commentaires s'appliquent ici que ceux décrits ci-dessus pour le flux de travail mono.

Modèle de gouttelettes CFA

Le choix des noyaux de drizzle est le même pour le drizzle CFA que pour le drizzle mono. Notez que les noyaux qui sont particulièrement enclins à générer des pixels nuls peuvent s'avérer délicats lorsqu'ils sont utilisés pour le drizzle CFA. Si vous avez des dizaines de milliers d'images comme dans une vidéo planétaire, turbo peut fonctionner correctement (et sera rapide !), mais pour des séquences de ciel profond avec un plus petit nombre d'images, il est recommandé de s'en tenir aux noyaux *carré* ou *Gaussien* (et gardez à l'esprit, comme mentionné ci-dessus, que Gaussian ne préserve pas le flux, donc si vous avez l'intention de faire quoi que ce soit impliquant des techniques photométriques, *carré* est préférable).

Pondération initiale des pixels CFA

Comme pour le drizzle mono, un master flat peut être spécifié. Pour activer le master flat, cochez la case*Inclure le master flatdans la pondération initiale des pixels*.

Avertissement

Le master flat doit être défini dans l'onglet*Calibration* !

C'est parti pour le Bayer Drizzle !

Once all the options are set, click the *Go register* button.

Empilez vos données CFA

Vous pouvez maintenant empiler votre séquence drizzle comme d'habitude, en notant les mêmes commentaires sur le rejet que pour le drizzle mono (ceux-ci peuvent être plus apparents avec le drizzle CFA si vous avez une couverture inadéquate pour soutenir certains des algorithmes de rejet des valeurs aberrantes, en raison de la plus grande rareté des pixels d'entrée dans chaque canal).

Astuce

Si vous utilisez le drizzle pour améliorer la résolution de vos données CFA, il est possible que vous soyez déçu lorsque vous comparez les résultats avec des images dématricées empilées. Il y a généralement un gain, mais il peut être marginal (par exemple, une amélioration de quelques pour cent de la fwhm) et ne sera généralement pas aussi impressionnant que le gain en résolution obtenu avec le drizzle des données mono sous-échantillonnées.

La raison en est que le dématriçage rétablit déjà une partie de la résolution perdue. Les divers algorithmes de dématriçage fonctionnent différemment, mais ils s'appuient généralement tous sur les principes de corrélation spatiale et spectrale pour déduire une partie de la résolution manquante dans un canal sur la base des informations obtenues dans les autres canaux. [Losson2010]

La véritable raison d'appliquer un Drizzle CFA est que le résultat a un bruit plus propre. Il semble moins "granuleux" (c'est-à-dire qu'il n'a pas la structure que l'on peut voir dans l'arrière-plan d'un empilement CFA dématricé typique) et est donc plus facile à réduire à l'aide de techniques de réduction du bruit et donne des données plus cohérentes pour les applications photométriques telles que l'étalonnage des couleurs. Lorsqu'on l'étire fortement pour faire ressortir des éléments peu visibles juste au-dessus du niveau du fond de ciel, le fond de ciel obtenu parait plus naturel.

Comparaison du Bayer Drizzle

L'animation ci-dessous montre une comparaison entre le drizzle CFA avec deux fractions de pixels différentes et deux des algorithmes classiques de dématriçage.

Fig. 37 – Comparaison de l'algorithme CFA drizzle (appelé ici Bayer Drizzle) avec les algorithmes classiques de dématriçage

- VNG est fourni comme référence de base : notez l'artefact de couleur autour des étoiles les plus brillantes.
- RCD est très efficace pour les objets ronds tels que les étoiles.
- Bayer Drizzle 1.0 donne des résultats très proches de RCD mais avec un meilleur bruit et un meilleur fond de ciel
- Bayer Drizzle 0.5 offre une meilleure résolution au prix d'un bruit plus important. Le compromis que la fraction de pixel permet d'obtenir entre la résolution et le bruit est évident. Avec une fraction de pixels plus petite, le drizzle CFA a besoin de plus de données pour obtenir la même performance en matière de bruit.

9.4.7 Quelques problèmes courants

> **Astuce**
>
> **NE PANIQUEZ PAS** — les résultats suivants peuvent sembler un peu bizarres lorsque vous regardez une image drizzle individuelle, mais ce n'est pas un bug — l'algorithme fonctionne comme prévu. Dans la plupart des cas, cela se résout naturellement lors de l'empilement. Dans les autres cas, ils peuvent être résolus en modifiant les paramètres du drizzle ou en incluant plus d'images.

Motifs moirés

En raison de la nature de l'algorithme de drizzle, lors de la mise à l'échelle, certains pixels de sortie peuvent ne pas recevoir de valeur. Ces pixels sont appelés "pixels nuls" et ont une valeur nulle. Certains noyaux compensent ce phénomène en limitant la fraction de pixel, de sorte que tous les pixels de sortie reçoivent une entrée, mais d'autres ne le font pas.

Les pixels de sortie qui ne reçoivent pas de valeurs sont noirs : comme ils se produisent généralement dans des motifs basés sur la géométrie de la transformation de l'image d'entrée, ils ressemblent typiquement à des motifs moirés, comme illustré ci-dessous :

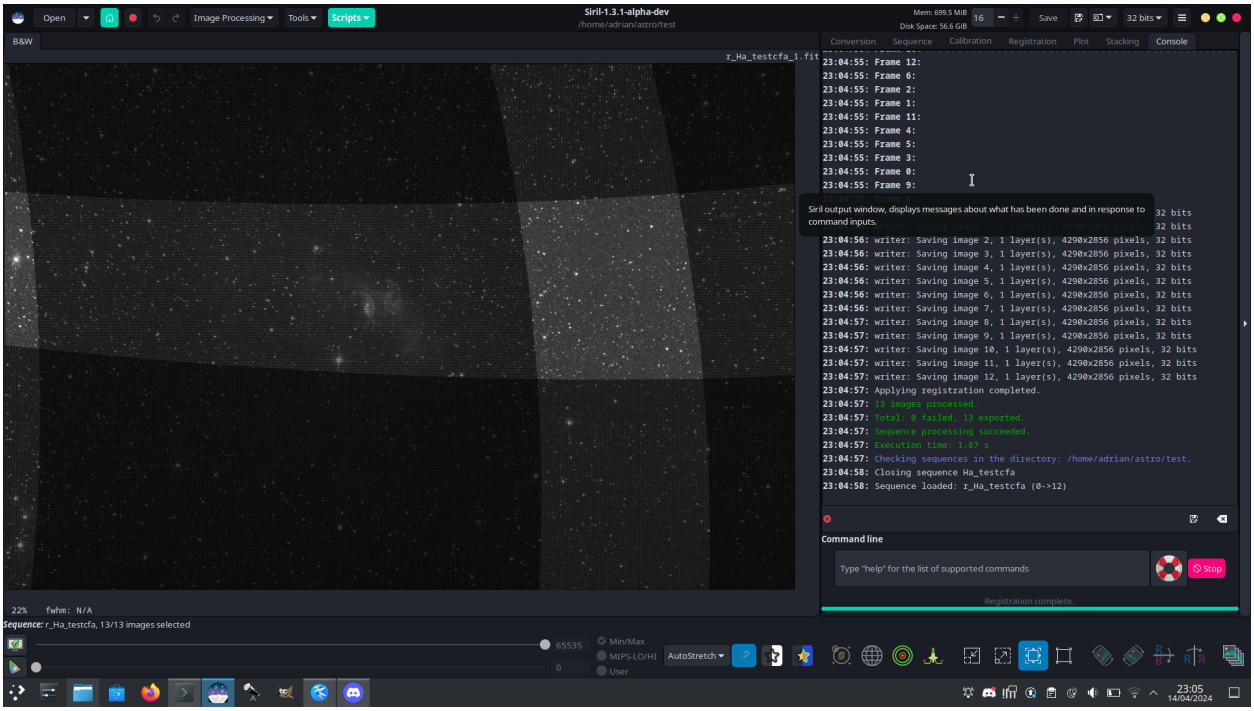


Fig. 38 – Représentation des motifs résultant de pixels nuls dans une image drizzle

Ne vous inquiétez pas pour cela ! Siril ignore les pixels qui sont exactement à 0 dans l'empilement, donc tant que vous avez suffisamment d'images en entrée et que les positions des pixels sont convenablement dispersées, tous les pixels recevront une couverture de suffisamment de pixels et la pile de sortie sera correcte. Cependant, si vous empilez avec un nombre inférieur d'images en entrée et que cela pose des problèmes, essayez un autre noyau de drizzle. Voici précisément la même image drizzle avec exactement la même échelle et la même fraction de pixel, mais avec le noyau carré au lieu du noyau turbo. Le résultat est différent et les motifs ne sont plus évidents.

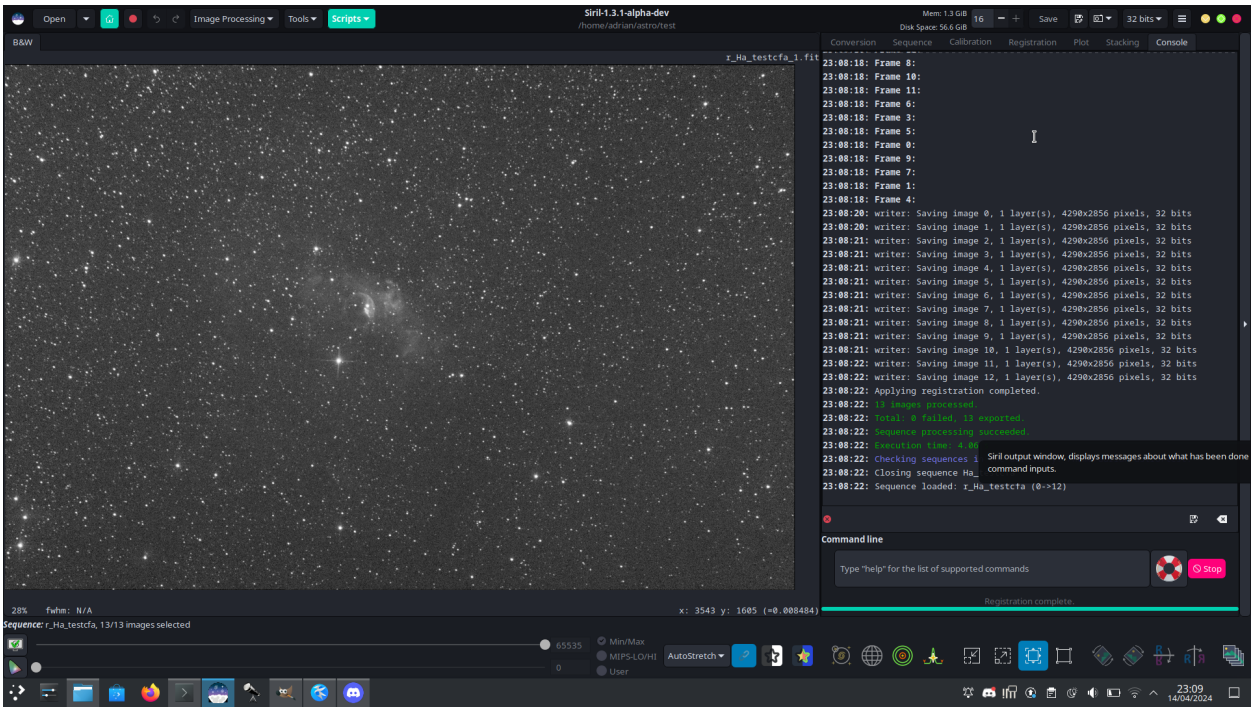


FIG. 39 – L'utilisation d'un noyau de drizzle différent permet d'éliminer les motifs de pixels nuls

Stacks de patchs

Lors de l'empilement de données "drizzle", si le nombre de pixels nuls est trop élevé, le résultat final peut présenter un aspect irrégulier :

Cela se produit généralement avec les noyaux ponctuels, turbo ou Lanczos. Vous pouvez y remédier en utilisant les noyaux carrés ou gaussiens ou en ayant plus d'images d'entrée.

Commandes

Ligne de commande Siril

```
seqapplyreg sequencename [-prefix=] [-scale=] [-layer=] [-framing=]
seqapplyreg sequencename ... [-interp=] [-noclamp]
seqapplyreg sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
seqapplyreg sequencename ... [-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-
↪filter-round=value[%|k]] [-filter-bkg=value[%|k]] [-filter-nbstars=value[%|k]] [-
↪filter-quality=value[%|k]] [-filter-incl[uded]]
```

Applique des transformations géométriques aux images de la séquence donnée en argument afin qu'elles puissent être superposées à l'image de référence, en utilisant les données d'alignement précédemment calculées (voir REGISTER).

Le nom de la séquence de sortie commence par le préfixe **"r_"** sauf indication contraire spécifié par l'option **-prefix=**.

L'alignement est réalisé sur la première couche pour laquelle les données existe pour une image RVB sauf si spécifié par l'option **-layer=** (0, 1 ou 2 pour R, V et B respectivement).

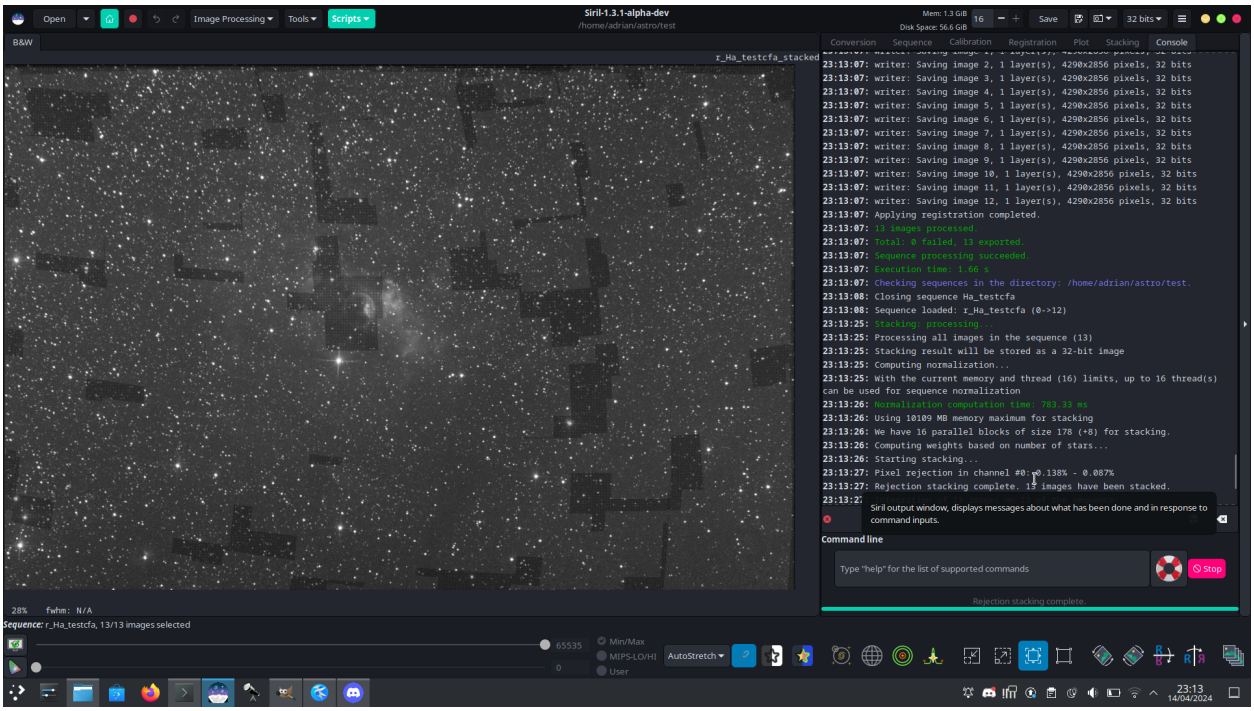


FIG. 40 – Aspect disparate typique d'un empilement déformé avec trop de pixels nuls / pas assez d'images

The output images can be rescaled by passing a **-scale=** argument with a float value between 0.1 and 3.

Le cadrage automatique de la séquence de sortie peut être spécifié en utilisant le mot-clé **-framing=** suivi de l'une des méthodes de la liste { current | min | max | cog } :

-framing=max (bounding box) will project each image and compute its shift wrt. reference image. The resulting sequence can then be stacked using option **-maximize** of STACK command which will create the full image encompassing all images of the sequence.

-framing=min (zone commune) recadre chaque image dans la zone qu'elle a en commun avec toutes les images de la séquence.

-framing=cog détermine la meilleur position de cadrage comme centre de gravité (cog) de toutes les images.

Option d'interpolation de l'image :

Par défaut, les transformations sont appliquées pour aligner les images à l'aide de l'interpolation.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.

La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**.

Options de drizzle pour les images :

Sinon, les images peuvent être exportées à l'aide de l'algorithme HST drizzle en passant l'argument **-drizzle** qui peut prendre les options supplémentaires suivantes :

-pixfrac= défini la fraction de pixel (par défaut =1.0).

L'argument **-kernel=** défini le noyau de drizzle et doit être suivi par l'un des arguments suivant : **point**, **turbo**, **square**, **gaussian**, **lanczos2** or **lanczos3**. La valeur par défaut est **square**.

L'argument **-flat=** spécifie un flat maître pour pondérer les pixels d'entrée drizzlés (la valeur par défaut est sans flat).

Filtrage des images :
Les images qui vont être aligner peuvent être sélectionnées en utilisant des filtres, tel que celles sélectionné ou avec la meilleur FWHM, avec l'une des option **-filter-***.

Links : *register*, *stack*

Note that the introduction of true drizzle has necessitated some changes to existing command arguments for clarity.

register and *seqapplyreg* have a new argument **-drizzle** which, together with some related arguments, activates true drizzle.

Références

9.5 Empilement

La dernière étape de prétraitement à effectuer avec Siril consiste à empiler les images. L'empilement d'images est une technique utilisée en astrophotographie pour améliorer la qualité et le détail d'une image en combinant plusieurs photographies en une seule image composite. Le processus consiste à prendre plusieurs images du même objet, puis à les aligner et à en faire la moyenne afin de réduire le bruit et d'augmenter le rapport signal/bruit. Il en résulte une image finale moins bruitée, plus détaillée et avec une plus grande plage dynamique qu'une simple exposition.

9.5.1 Méthodes d'empilement

Empilement par somme

Il s'agit de l'algorithme le plus simple : chaque pixel de la pile est additionné. L'augmentation du rapport signal/bruit (RSB) est proportionnelle à $\sqrt{N}$, où N est le nombre d'images. En raison de l'absence de normalisation et de rejet, cette méthode ne doit être utilisée que pour le traitement des planètes.

Pour les images d'entrée à 8 ou 16 bits par canal, la somme est effectuée dans un entier de 64 bits avant d'être normalisée à la valeur maximale du pixel et sauvegardée comme un entier non signé de 16 bits ou dans un nombre à virgule flottante de 32 bits.

Cette méthode d'empilement devrait être utilisée pour les images d'entrée de 8 bits, car elle augmente la dynamique des images tout en les empilant, ce qui permet de discerner les caractéristiques. L'empilement avec une méthode moyenne ou médiane d'une telle séquence ne ferait que diminuer le bruit mais n'améliorerait pas la dynamique de l'image, le résultat étant toujours d'une profondeur de 8 bits.

Empilement par moyenne avec rejet des pixels déviants

Cette méthode d'empilement calcule une moyenne des pixels d'une pile après avoir exclu les pixels déviants et une effectue une normalisation facultative des images par rapport à l'image de référence. Comme pour l'empilement par somme, l'amélioration du rapport signal/bruit est proportionnelle à $\sqrt{N}$. Il existe plusieurs façons de normaliser les images et plusieurs façons de détecter et de remplacer ou d'exclure les pixels déviants, comme expliqué ci-dessous.

Avertissement

Certains systèmes d'exploitation limitent le nombre d'images pouvant être ouvertes simultanément, ce qui est nécessaire pour les méthodes d'emnepilement par médiane ou moyen. Pour Windows, la limite est de 2048 images.

Si vous avez beaucoup d'images, vous devez utiliser un autre type de séquence, décrit *ici*.

FIG. 41 – Panneau d'empilement par rejet de pixel

Méthodes de rejet

- **Percentile Clipping** : C'est un algorithme de rejet de pixels en une étape, idéal pour les petits ensemble d'images (jusqu'à 6 images).
- **Sigma Clipping** : Il s'agit d'un algorithme itératif qui rejette les pixels dont la distance par rapport à la médiane est supérieure à deux valeurs données en unités sigma ($sigma$ low, σ high).
- **MAD Clipping** : Il s'agit d'un algorithme itératif qui fonctionne comme le Sigma Clipping, sauf que l'estimateur utilisé est l'écart absolu médian (MAD). Cet algorithme est généralement utilisé pour le traitement d'images infrarouges bruitées.
- **Sigma Clipping Médian** : Il s'agit du même algorithme que le Sigma Clipping, sauf que les pixels rejetés sont remplacés par la valeur médiane de la pile.
- **Winsorized Sigma Clipping** : Cette méthode est très similaire à la méthode Sigma Clipping, sauf qu'elle est censée être plus robuste pour la détection des valeurs aberrantes, voir le travail de Huber [Peter2009].
- **Test de déviation extrême généralisé de Student** [Rosner1983] : Il s'agit d'une généralisation du test de Grubbs qui est utilisé pour détecter une ou plusieurs valeurs aberrantes dans un ensemble de données univariées qui suit une distribution approximativement normale. Cet algorithme montre d'excellentes performances avec un grand ensemble de données de plus de 50 images.
- **Linear Fit Clipping** [ConejeroPI] : Il ajuste la meilleure ligne droite ($y = ax + b$) de la pile de pixels et rejette les valeurs aberrantes. Cet algorithme fonctionne très bien avec les grandes piles et les images contenant des gradients de ciel avec des distributions spatiales et des orientations différentes.

Notez que pour les images CFA drizzle, vous pouvez avoir des difficultés à obtenir un rejet satisfaisant dans les canaux rouge et bleu avec certaines méthodes. Dans ce cas, la méthode de clipping MAD semble être la plus efficace.

Carte de rejet

L'option *Créer cartes de rejet* calcule et crée des cartes de rejet pendant l'empilement. Ce sont des images montrant combien d'images ont été rejetées pour chaque pixel de l'image résultat, divisé par le nombre d'images empilées. Si *Fusionner L+H* est coché, Siril ne crée qu'une seule carte de rejet qui sera la somme des cartes basse et haute.

Filtrage/poids des images

La pondération permet d'attribuer un poids statistique à chaque image. Ainsi, les images considérées comme les meilleures contribueront davantage que celles considérées comme les moins bonnes. Quatre méthodes de pondération sont disponibles :

- **Nombre d'étoiles** donne des poids aux images individuelles en fonction du nombre d'étoiles calculé lors de l'étape d'alignement.

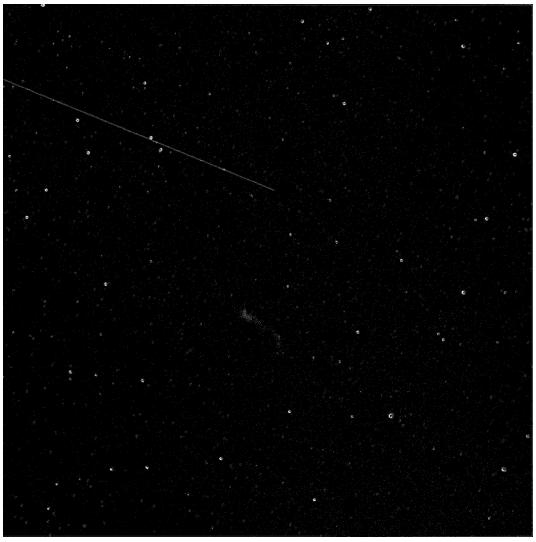


FIG. 42 – Exemple de carte de rejet (L+H). On voit très clairement la trace d'un satellite qui a été retiré.

— **Weighted FWHM** pondère les images individuelles sur la base de la wFWHM calculée lors de l'étape d'alignement. Il s'agit d'une FWHM pondérée par le nombre d'étoiles dans l'image. Pour une même mesure de FWHM, une image avec plus d'étoiles aura une meilleure wFWHM qu'une image avec moins d'étoiles.
— **Bruit** pondère les images individuelles en fonction des valeurs de bruit de fond.
— **Nombre d'images** pondère les images individuelles en fonction de leur temps d'intégration.

Assemblage d'images

Available since Siril 1.3, this submenu allows to perform a smoother stitching between images with little overlaps. It is the first step of mosaic stitching which we plan to expand further in next releases.

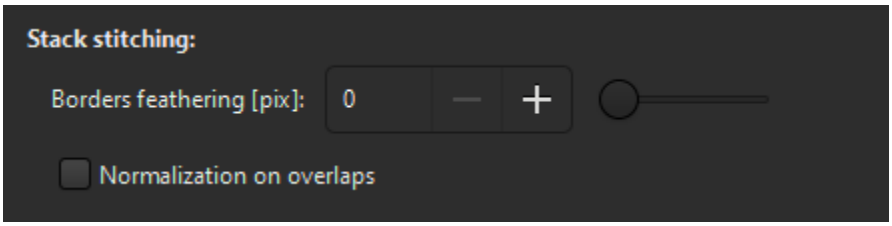


FIG. 43 – Panneau d'assemblage d'empilements

— *Chevauchement des contours* applique un masque de lissage à chaque image lors de l'empilement. Le lissage s'effectue sur une distance de X pixels, X correspondant à la valeur que vous définissez via cette interface. Les masques sont mis en cache dans le sous-dossier `./cache`.

FIG. 44 – Effet de chevauchement des contours appliqué à une mosaïque 3x2 (négatif). Images par G. Attard

— *Normalization on overlaps* will compute the normalization factors on the images overlaps instead of whole images. This is useful when images have little overlap and when the nature of the data in each tile is very different accross all images (some images with mostly nebulosity and others with mainly black sky). As it is longer to compute than regular normalization, you should try without ticking this option first. This feature is intended for stitching stacked mosaic tiles so it will throw a warning if you try to compute it for a long sequence (*i.e.* for subs acquired with a smart telescope in mosaic mode). It is only available if *maximize framing* is enabled.

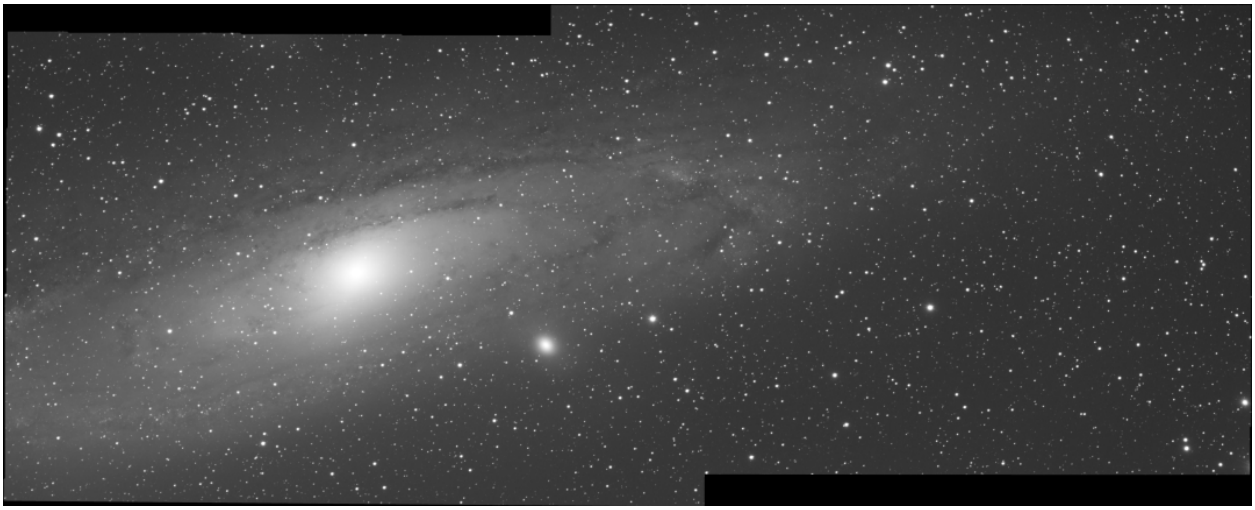


FIG. 45 – Global vs overlap normalization on a 2x1 mosaic. Images courtesy of D. Huber

Théorie

The normalization on overlaps has been derived from the algorithm for gain compensation from [Brown2007]. The full write-up of the modifications is described in the contributors documentation.

Empilement médian

Cette méthode est principalement utilisée pour l'empilement des darks/flats/offsets. La valeur médiane des pixels de la pile est calculée pour chaque pixel.

L'augmentation du rapport signal/bruit est proportionnelle à $0.8\sqrt{N}$ et est donc moins bonne que l'empilement par moyenne qui est généralement préféré.

Empilement par pixel maximum

Cet algorithme est principalement utilisé pour construire des images de traînées d'étoiles à longue exposition. Les pixels de l'image sont remplacés par des pixels aux mêmes coordonnées si l'intensité est plus grande.

Empilement du pixel minimum

Cet algorithme est principalement utilisé pour recadrer une séquence en supprimant les bordures noires. Les pixels de l'image sont remplacés par des pixels aux mêmes coordonnées si l'intensité est plus faible.

9.5.2 Méthodes de normalisation en entrée

La normalisation ajuste les niveaux de chaque image par rapport à l'image de référence. Cette opération est particulièrement utile pour l'empilement par moyenne avec rejet des pixels déviants, car il n'est pas très utile de rejeter des pixels si les images présentent des différences de niveau. Ces différences peuvent être dues à la nébulosité, au gradient de lumière causé par la lune ou les lumières de la ville, à la variation de la température du capteur, etc.

Cela tend à améliorer le rapport signal/bruit et c'est donc l'option utilisée par défaut avec la normalisation additive.

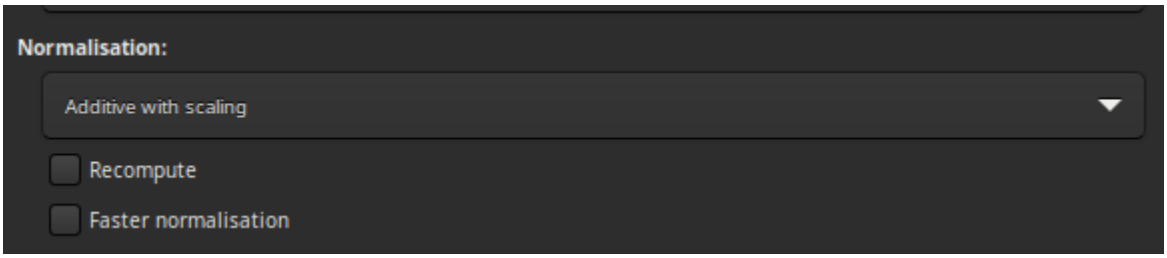


Fig. 46 – Panneau de normalisation des entrées d'empilement

Si l'un de ces cinq éléments est sélectionné, un processus de normalisation sera appliqué à toutes les images d'entrée avant l'empilement.

— La normalisation fait correspondre l'arrière-plan moyen de toutes les images d'entrée, et la normalisation est appliquée par multiplication ou addition. Il faut garder à l'esprit que les deux processus aboutissent généralement à des résultats similaires, mais que la normalisation multiplicative est préférable pour les images qui seront utilisées pour la multiplication ou la division en tant qu'images flats.

— La mise à l'échelle gère la dispersion en pondérant toutes les images. Cette opération tend à améliorer le rapport signal/bruit et par conséquent cette option est active par défaut avec a normalisation additive.

Normalisation	Opération	Cas pratique
Aucun	Aucune normalisation n'est appliquée.	images dark/bias
Additive	Les valeurs moyennes du fond de ciel seront alignées par l'application d'opérations additives.	
Multiplicative	La division sera utilisée pour aligner les valeurs moyennes du fond de ciel.	images flat
Additive + Mise à l'échelle	En combinaison avec la correspondance additive du fond de ciel, les images seront mises à l'échelle pour obtenir la correspondance de la dispersion.	Images cible
Multiplicative + Mise à l'échelle	En combinaison avec la correspondance du fond de ciel par division, les images seront mises à l'échelle pour obtenir la correspondance de la dispersion.	

Note

Les masters offset et dark ne doivent pas être traités avec la normalisation. Cependant, la normalisation multiplicative est recommandée pour les images flats.

Gardez à l'esprit que les deux processus aboutissent généralement à des résultats similaires, mais que la normalisation multiplicative est préférable pour les images qui seront utilisées pour la multiplication ou la division en tant qu'images flats.

Comme l'étape de calcul de la normalisation est généralement longue, puisqu'elle nécessite de déterminer toutes les statistiques de l'image, les résultats sont stockés dans le fichier `seq`. De cette façon, si l'utilisateur veut faire un autre empilement en changeant les paramètres de rejet, il sera exécuté plus rapidement. L'option *Recalculer* permet de forcer le recalcul de la normalisation.

Par défaut, Siril utilise des estimateurs IKSS de la position et de l'échelle pour calculer la normalisation. Pour les longues séquences, le calcul de ces estimateurs peut être assez intensif. Pour de tels cas, vous pouvez opter pour des estimateurs plus rapides (basés sur la médiane et l'écart absolu à la médiane) avec l'option *Normalisation plus rapide*. Bien qu'ils soient moins résistants aux valeurs aberrantes dans chaque image, ils peuvent tout de même donner un résultat satisfaisant par rapport à l'absence de normalisation.

9.5.3 Rejet des images

Il est également possible de rejeter un certain nombre d'images afin de ne sélectionner que les meilleures. Cela peut être très utile pour les techniques Lucky DSO où le nombre d'images dans une séquence est très élevé. On peut choisir entre % et k-σ pour soit conserver un pourcentage donné d'images, soit calculer le seuil admissible en utilisant le rejet k-σ.

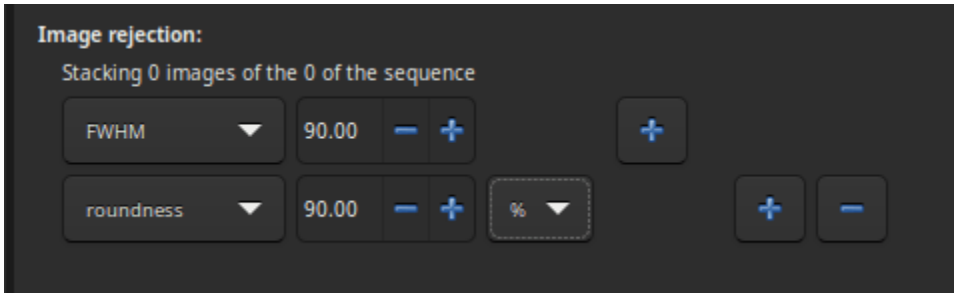


Fig. 47 – Panneau de rejet des images de l'empilement

Plusieurs critères sont disponibles :

- **toutes** : toutes les images de la séquence sont utilisées pour l'empilement.
- **sélectionnées** : utilise seulement les images qui n'ont pas été désélectionnées dans la séquence.
- **FWHM** : images avec le meilleur calcul FWHM (alignement basé sur les étoiles seulement).
- **FWHM pondérée** : c'est une amélioration de la simple FWHM. Cela permet d'exclure plus d'images suspectes en utilisant le nombre d'étoiles détectées par rapport à l'image de référence (alignement basé sur les étoiles seulement).
- **rondeur** : images avec la meilleure rondeur en étoile (alignement basé sur les étoiles seulement).
- **fond de ciel** : images avec les valeurs de fond du ciel les plus faibles (alignement basé sur les étoiles seulement).
- **nb étoiles** : images avec le meilleur nombres d'étoiles détectées alignement basé sur les étoiles seulement).
- **qualité** : images avec les meilleurs qualités (alignements planétaires DFT ou Kombat).

9.5.4 Résultat d'empilement

- Si l'option *Normalisation en sortie* est cochée, l'image finale sera normalisée dans l'intervalle [0, 1] si vous travaillez en précision 32 bits, ou dans l'intervalle [0, 65535] sinon.

> **Avertissement**
>
> Cette option ne doit pas être cochée pour l'empilement des maîtres.

- Si l'option *Égalisation RVB* est cochée, les canaux de l'image finale seront égalisés (uniquement pour les images en couleur).
- If *Maximize framing* is checked, the output image will encompass all the images. Note this option is forced if the images have different sizes.
- If *Interpolation upscaling x2* is ticked, the images will be upscaled during stacking. Note this option is not always available.
- Selecting *Force 32b* forces the stacked image to be saved as a float image irrespective of the bitdepth set in the Preferences.
- Le résultat de l'empilement est enregistré sous le nom donné dans le champ texte. Il est possible d'utiliser *l'interprétation d'entêtes* pour construire le nom de fichier. Cocher la case *Écraser* permet au nouveau fichier créé d'écraser l'ancien s'il existe. Si ce dernier n'est pas coché mais qu'une image portant le même nom existe déjà, alors aucun nouveau fichier n'est créé.

9.5.5 Références

CHAPITRE 10

Traitement

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 menus and sub-menus, and in this documentation too, by theme.

10.1 Étirement de l'image

Les images sont stockées sous forme de valeurs de pixels qui proviennent de l'appareil photo selon une loi quasi-linéaire, ce qui signifie que pour les zones du ciel qui ne présentent aucune caractéristique visible, la valeur du pixel sera proche de zéro, mais pour les objets brillants comme les étoiles, elle sera proche d'une valeur maximale en fonction de l'exposition et du gain. Entre les deux, si une nébuleuse a une magnitude de surface égale à la moitié de celle d'une étoile, elle aura des valeurs de pixels égales à la moitié de celles de l'étoile, et ainsi de suite. C'est ce que nous appelons le mode linéaire des pixels.

L'œil humain ne voit pas les photons de cette manière. Il amplifie les zones sombres, de sorte qu'un objet dix fois moins lumineux qu'un autre semble deux fois moins lumineux. Pour les images d'astronomie, nous affichons généralement les images avec une mise à l'échelle similaire des valeurs de pixels (voir les modes d'affichage de l'interface graphique).

Mais ce n'est qu'une astuce d'affichage, utilisant une fonction de transfert d'écran, pour rendre les valeurs de pixels de l'image intacte en images de meilleure qualité.

L'étirement d'image consiste à faire quelque chose de similaire, mais en modifiant les valeurs des pixels des images au lieu de simplement modifier leur rendu. Siril dispose de trois outils principaux pour y parvenir.

10.1.1 Transformation asinh

La transformation asinh, ou sinus hyperbolique inverse, modifie les valeurs des pixels de l'image d'une manière similaire à ce que l'on peut voir avec la fonction asinh de mise à l'échelle des pixels de l'écran, qui est paramétrée par les curseurs de coupure des valeurs basses et hautes. Ici, les paramètres sont le facteur d'étirement et la valeur du point noir.

Pour les images monochromes, la valeurs des pixels est modifiée en utilisant la fonction suivante :

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

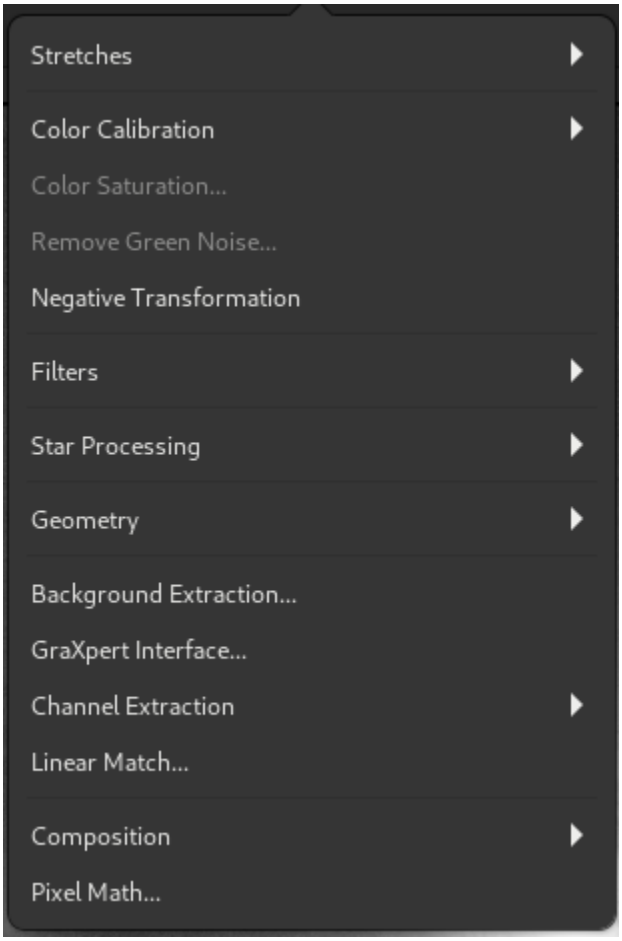


FIG. 1 – Menu traitement d'image

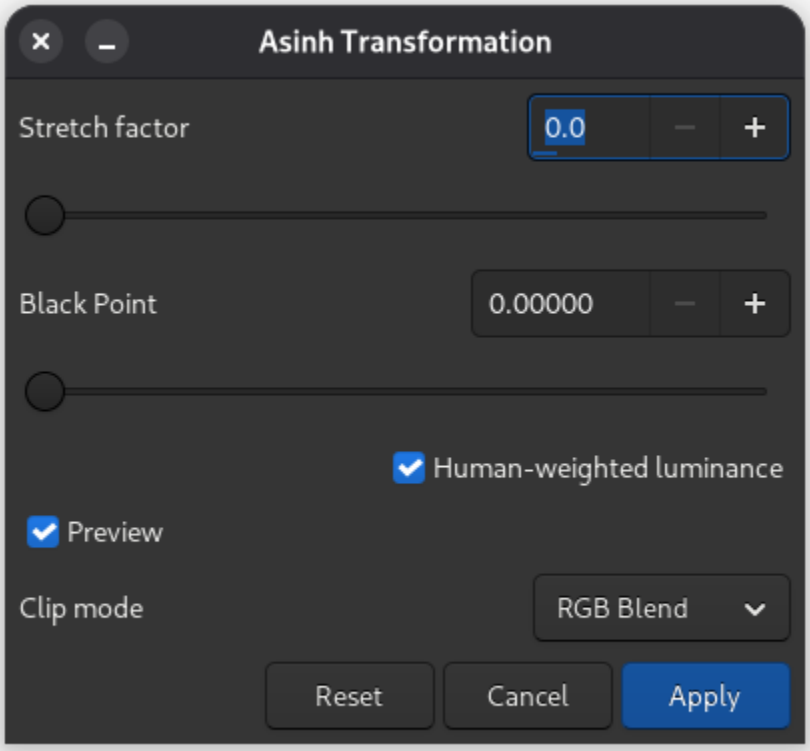


FIG. 2 – Boite de dialogue de la transformation asinh

Pour les images en couleur, la fonction devient :

$$\text{pixel} = \frac{(\text{original} - \text{blackpoint}) \times \text{asinh}(\text{rgb_original} \times \text{stretch})}{\text{rgb_original} \times \text{asinh}(\text{stretch})}$$

où rgb_original est calculé en utilisant les valeurs des pixels des trois canaux.

Il est également possible de sélectionner un mode de clipping.

Théorie

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 was developed by the authors of the original GHSastro tool : the same algorithms are available in the Siril implementation and this is the default clipping mode for stretches that require handling of clipping. 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

$k \times r' + (1 - k) \times r''1$;

$k \times g' + (1 - k) \times g''1$;

et

$k \times b' + (1 - k) \times b''1$

Les valeurs transformées sont ensuite calculées comme suit

$(k \times r' + (1 - k) \times r'', k \times g' + (1 - k) \times g'', k \times b' + (1 - k) \times b'')$

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

Other choices of clipping algorithm are available :

- *Clip* - this clipping mode just allows any colour components that clip to clip, but restricts them to values in the range 0.0 to 1.0. This may suffer from coloured artefacts such as fringes around nearly-saturated stars, but it is extremely quick to calculate.
- *Rescale* - this clipping mode checks the R, G and B components of each pixel and if any are > 1.0 it rescales the pixel so that no components are clipped. This method is prone to artefacts and is mainly included for feature equivalence with the GHSastro plugin. It is quick to compute.
- *Global Rescale* - this clipping mode behaves similarly to *Rescale* except that the scaling is computed globally instead of per-pixel. This avoids the kind of artefacts that Rescale can produce, but has a bigger impact on overall image brightness. This is faster than *RGB blending* to compute but slower than *Clip* or *Rescale*.

When the Use Human-weighted Luminance option is not ticked, rgb_original is the mean of the three pixel values ; when it is set, weighting 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 human perceptual color balance.

Ligne de commande Siril

```
asinh [-human] stretch { [offset] [-clipmode=] }
```

Etire l'image pour montrer les objets peu lumineux en utilisant une transformation hyperbolique arcsin. L'argument obligatoire **stretch**, généralement compris entre 1 et 1000, indique l'intensité de l'étirement. Le point noir peut être décalé en fournissant un argument optionnel **offset** dans la valeur normalisée du pixel [0, 1]. Enfin,

l'option **-human** permet d'utiliser les facteurs d'efficacité lumineuse de l'œil humain pour déterminer la luminance utilisée pour calculer la valeur d'étirement pour chaque pixel, au lieu de la simple moyenne des valeurs de pixel des canaux. Cette méthode d'étirement préserve la luminosité de l'espace colorimétrique La*b*. Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

10.1.2 Transformation de la fonction de transfert de la tonalité moyenne (MTF)

La MTF est l'un des outils les plus puissants pour étirer l'image. Elle peut être facilement automatisée et c'est pourquoi la vue auto-étirée l'utilise.

L'outil se présente sous la forme d'un histogramme avec 3 curseurs (en forme de triangle placé en dessous) qu'il faut déplacer pour transformer l'image. Le triangle de gauche représente le signal des ombres, celui de droite les hautes lumières et enfin, celui du milieu le paramètre de balance des tons moyens. Les valeurs de ces curseurs sont affichées sous l'histogramme, à gauche, et peuvent être modifiées directement à la main. Ci-contre, le pourcentage de pixels écrêtés par la transformation : il est important de ne pas écrêter trop de pixels. Si seul le paramètre des tons moyens est modifié, aucun pixel ne peut être écrêté.

Théorie

La nouvelle valeur de pixel est calculé grâce à la fonction :

$$\mathrm{MTF}(x_p) = \frac{(m-1)x_p}{(2m-1)x_p - m}. \tag{10.1}$$

- Pour$x_p = 0$,MTF $= 0$,
- pour$x_p = m$,MTF $= 0.5$,
- pour$x_p = 1$,MTF $= 1$,

où x_p est la valeur du pixel défini comme

$$x_p = \frac{\text{original} - \text{shadows}}{\text{highlights} - \text{shadows}}. \tag{10.2}$$

Note

Il n'est généralement pas recommandé de modifier la valeur des hautes lumières, sinon elles seront saturées et l'information sera perdue.

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 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.

Astuce

When the button is pressed, the sliders and lo, mid and hi entries will become temporarily inactive. You have to apply the autostretch with the *Apply* button and the controls will reactivate. You can then apply adjustments as a

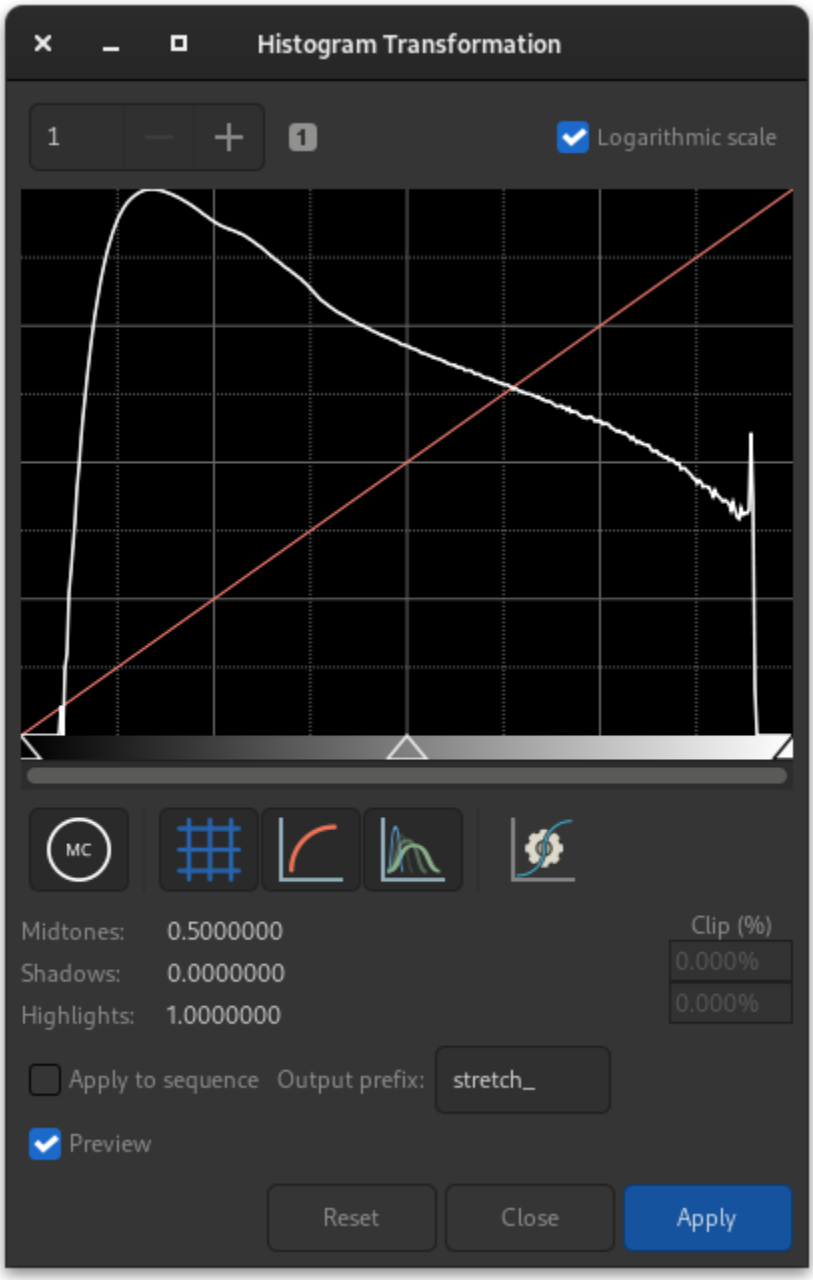


FIG. 3 – Boite de dialogue de la transformation de l'histogramme

second follow-up stretch. This behaviour avoids problems where the monitor ICC profile is different to the image ICC profile.

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 *here*. Finally a zoom in X is available. This is very useful when all the signal is concentrated on the left of the histogram.

Astuce

If a ROI is set, the MTF histogram preview will not update to show the impact of the stretch on the ROI. This is because that behaviour could be misleading : if the ROI is not typical of the image overall, adjusting the ROI histogram to a suitable level would result in a badly adjusted histogram for the overall image and potentially a burned-out or excessively dark look to the result. When in ROI mode the stretch parameters should be adjusted by eye. If it is desired to check the histogram for the stretch as applied to the image as a whole, the ROI should be cleared.

Ligne de commande Siril

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

Applique la fonction de transfert des demi-tons à l'image actuellement chargée.

Trois paramètres sont nécessaires, **low**, **midtones** et **high** où le paramètre de balance des tons moyens définit un étirement non linéaire de l'histogramme dans la plage [0,1]. Pour une détermination automatique des paramètres, voir AUTOSTRETCH.
Le paramètre optionnel **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, G, B, RG, RB ou GB. La valeur par défaut est tous les canaux

Liens : *autostretch*

Note

`mtf` est également une fonction qui peut être utilisée dans l'outil *PixelMath*.

Ligne de commande Siril

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

Etire automatiquement l'image actuellement chargée, avec des paramètres différents pour chaque canal (non lié) à moins que **-linked** ne soit fourni. Les arguments sont optionnels, **shadowclip** est le point d'écrêtage des ombres, mesuré en unités sigma à partir du pic de l'histogramme principal (la valeur par défaut est -2.8), **targetbg** est la valeur cible de l'arrière-plan, donnant une luminosité finale à l'image, plage [0, 1], la valeur par défaut est 0.25. Les valeurs par défaut sont celles utilisées dans le rendu Auto-Ajustement à partir de l'interface graphique.

N'utilisez pas la version non liée après l'étalonnage des couleurs, car elle modifierait la balance des blancs

Appliquer la transformation à la séquence

Cette transformation peut facilement être appliquée à une séquence. Il suffit de définir la transformation sur l'image chargée (avec une séquence déjà chargée), puis de cocher le bouton *Appliquer à la séquence* et de définir le préfixe de sortie de la nouvelle séquence (`stretch_` par défaut), ou d'utiliser la commande suivante :

Ligne de commande Siril

```
seqmtf sequencename low mid high [channels] [-prefix=]
```

Même commande que pour MTF mais pour la séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "mtf_", sauf indication contraire avec l'option **-prefix=**

Liens : *mtf*

10.1.3 Transformations hyperboliques généralisées (GHS)

Il s'agit de l'outil le plus performant et le plus moderne de Siril, mais aussi le plus complexe à apprendre. Un tutoriel très détaillé pour cet outil dans Siril a été écrit par les auteurs de cet algorithme : https://siril.org/tutorials/ghs. Nous nous contenterons ici de résumer le fonctionnement de base de cet outil.

En d'autres termes, le SGH est capable d'améliorer le contraste d'une gamme de niveaux de luminosité dans une image. Par exemple, si l'on souhaite mieux voir les détails dans la partie de luminosité moyenne à élevée d'une nébuleuse (qui est en général très faible dans une image d'astronomie), il est possible de ne sélectionner que cette plage pour l'étirement. L'outil est très efficace pour améliorer le contraste des objets principaux sans rendre les étoiles trop grosses. L'outil est basé sur une utilisation itérative, c'est-à-dire qu'il étire toutes les plages de luminosité de l'image l'une après l'autre, par petites touches.

Pour ce faire, l'outil s'appuie fortement sur l'affichage et l'interaction de l'histogramme, pour chaque canal de couleur. La fonction de transformation, en forme d'hyperbole ou de 'S', peut être modifiée en déplaçant son centre (le paramètre **SP** - point de symétrie), en aplatissant l'une ou l'autre de ses extrémités (avec les protections **ombre** et **haute lumière**), et bien sûr sa torsion (facteurs d'étirement **D** et d'étirement local **b**). La manipulation de ces paramètres sur une petite image (pour aller vite) avec une valeur **SP** de 0,5 vous aidera à comprendre leur effet.

Il y a deux opérations principales à effectuer à chaque itération : sélectionner la plage de lumières à modifier, et la modifier effectivement. La sélection de la plage est assez facile, il s'agit de trouver une valeur représentative (**SP**) et de définir la largeur de la plage (**b**). La définition de **SP** peut se faire de trois manières :

- en sélectionnant une zone de luminosité identique dans l'image et en cliquant sur le bouton du sélecteur
- cliquer directement sur l'histogramme avec un clic gauche (il est possible de zoomer dans l'histogramme en utilisant le bouton + en haut à gauche)
- en utilisant le curseur ou les boutons plus et moins qui lui sont associés ou directement la valeur.

La largeur de la plage dépend de l'étirement local. Une valeur élevée de **b** créera une petite plage et augmentera le contraste sur une petite plage de luminosité dans l'image.

La modification de l'histogramme une fois que l'emplacement du changement a été fixé est une opération plus complexe. L'un des objectifs fixés par les auteurs de l'algorithme est de faire en sorte que la vue logarithmique de l'histogramme (activée en cochant la case) soit aussi proche que possible d'une ligne décroissante. Pour ce faire, des bosses doivent être creusées et des vallées comblées. Voici un guide rapide des valeurs à utiliser en fonction de ce que l'on souhaite obtenir :

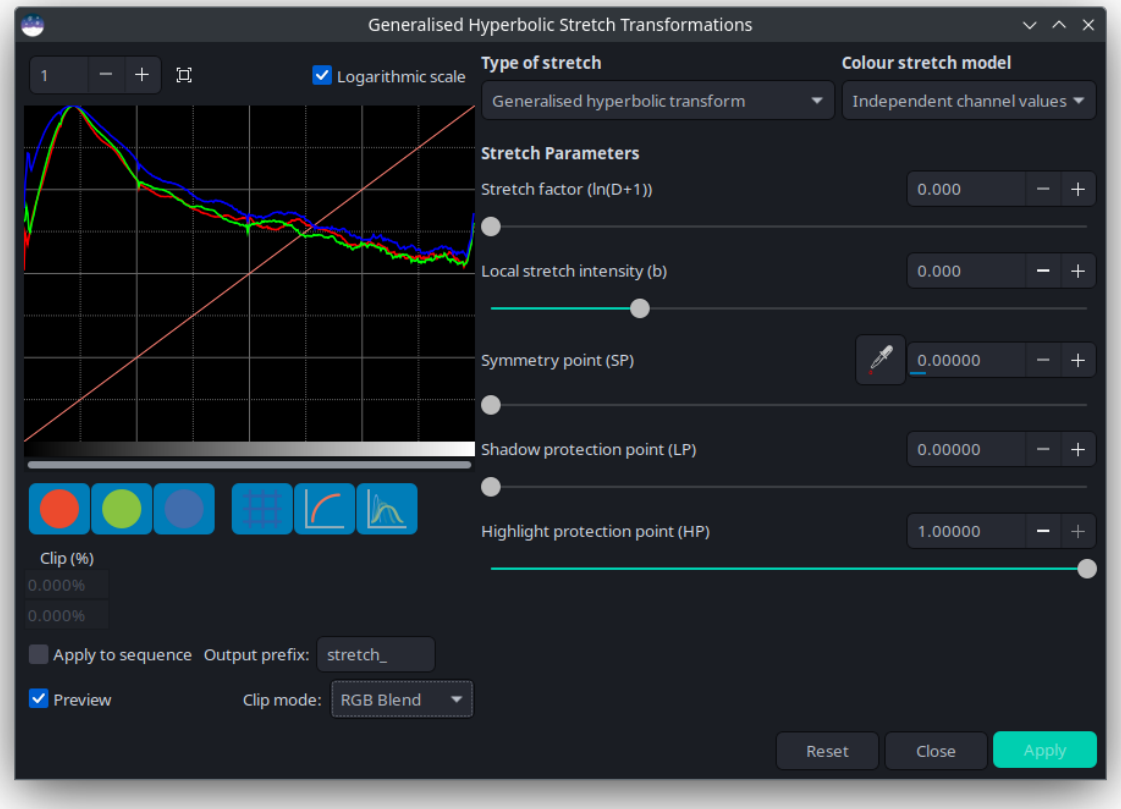


FIG. 4 – Boite de dialogue de la transformé hyperbolique généralisée

— **Étirement initial à partir de linéaire :** réglez **SP** légèrement à gauche du pic principal, modérez **b** à partir de 6, augmentez **D** légèrement pour commencer à voir l'objet principal. Ne pas trop étirer à ce stade comme le ferait un auto ajustement, sinon les étoiles deviendraient trop grosses (principale section du tutoriel pour cela).
— **Amélioration du contraste d'une plage ou remplissage d'une vallée :** réglez **SP** au centre de la vallée dans l'histogramme, réglez **b** aussi haut que la plage ou la vallée est étroite, diminuez **HP** pour préserver les étoiles, augmentez **D** lentement jusqu'à ce que l'amélioration apparaisse.
— **Diminuer le contraste d'une plage ou aplanir un pic :** La diminution d'un pic n'est pas facile à réaliser, mais elle est un effet secondaire du remplissage des vallées. Par exemple, la création d'un pic, ou le remplissage d'une vallée, diminuera ce qui se trouve à gauche de **SP**. Une autre possibilité consiste à utiliser la transformation inverse, à partir de la liste déroulante *Type d'étirement*, et une valeur **LP** élevée, et **HP** à 1.
— **déplacer la courbe vers la gauche, rendant l'image plus sombre :** souvent, si l'on étire tout l'histogramme, le pic se déplace vers la droite, rendant l'arrière-plan trop lumineux. Il existe un moyen simple de tout déplacer vers la gauche, en sélectionnant dans la liste déroulante *Type d'étirement* la dernière entrée, **Étirement linéaire (décalage BP)**. Il n'y a plus qu'un seul curseur à déplacer, contrôlant l'ampleur du déplacement.

Astuce

If a ROI is set, the GHT histogram preview will not update to show the impact of the stretch on the ROI. This is because that behaviour could be misleading : if the ROI is not typical of the image overall, adjusting the ROI histogram to a suitable level would result in a badly adjusted histogram for the overall image and potentially a burned-out or excessively dark look to the result. When in ROI mode the stretch parameters should be adjusted by eye. If it is desired to check the histogram for the stretch as applied to the image as a whole, the ROI should be cleared.

Certaines opérations sont également courantes pour les **images couleurs**, où l'on souhaite souvent obtenir une forme de courbe similaire pour les trois canaux, en travaillant sur chaque canal indépendamment en les désélectionnant à l'aide des trois cercles colorés situés sous la vue de l'histogramme :

— **déplacement du pic vers la droite :** un simple étirement avec une valeur **SP** à gauche du pic le fera en général, donc cela devrait être fait dans le cadre d'un étirement.
— **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 decreasing **HP**.
— **bouger tous les canaux ensemble :** une alternative d'étirement de la luminance existe, voir la combo box *Modèle d'étirement de couleur* en haut à droite de la fenêtre GHS, en utilisant l'une ou l'autre des valeurs d'étirement de la luminance permet d'étirer la luminance et de réappliquer les couleurs sur celle-ci au lieu d'étirer directement les trois canaux. Les modes de luminance permettent de mieux préserver les couleurs de l'image. Ces modes utilisent le même mode d'écrêtage du mélange RVB décrit ci-dessus pour éviter les artefacts d'écrêtage des canaux de couleur.
— **Saturation de l'image : ** Les transformations GHS peuvent être appliquées au canal de saturation de l'image en sélectionnant l'option Saturation dans la liste déroulante *Modèle d'étirement des couleurs*. Lorsque ce mode est sélectionné, les histogrammes de saturation avant et après l'étirement sont affichés en jaune. Toutes les options du GHS sont disponibles et ce mode permet un ajustement très ciblé du canal de saturation de l'image. Une méthode simple pour augmenter la saturation dans les régions relativement non saturées tout en évitant la sursaturation consiste à utiliser un étirement par **transformation hyperbolique généralisée inverse** avec **SP** fixé à environ 0,5 et **HP** abaissé suffisamment pour aplatir l'extrémité supérieure de l'histogramme de saturation.

In order to control partially-clipped highlights, the GHS tool makes available the same range of clipping modes as the asinh stretch. Details can be found *here*.

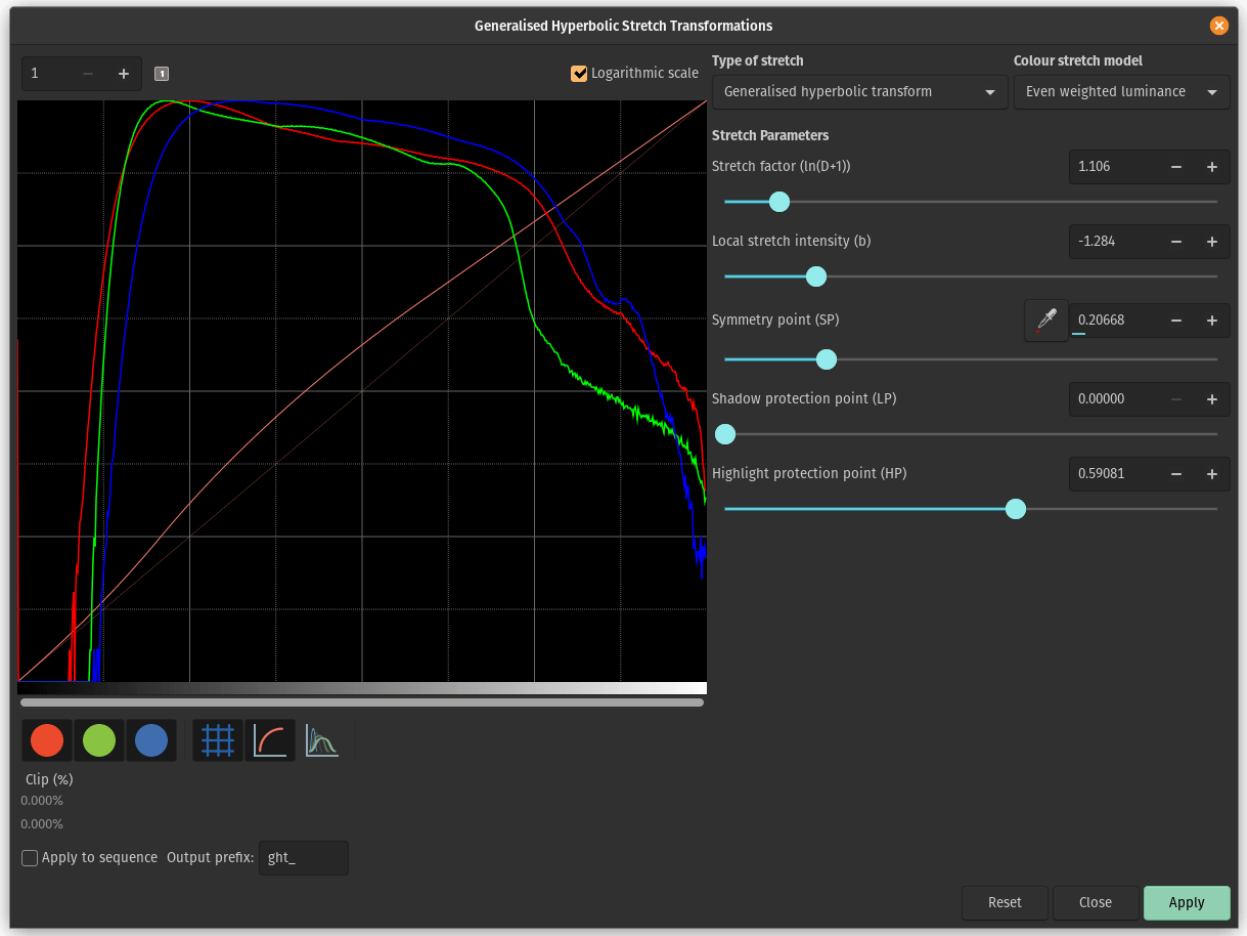


FIG. 5 – La transformation hyperbolique généralisée sur une image couleur

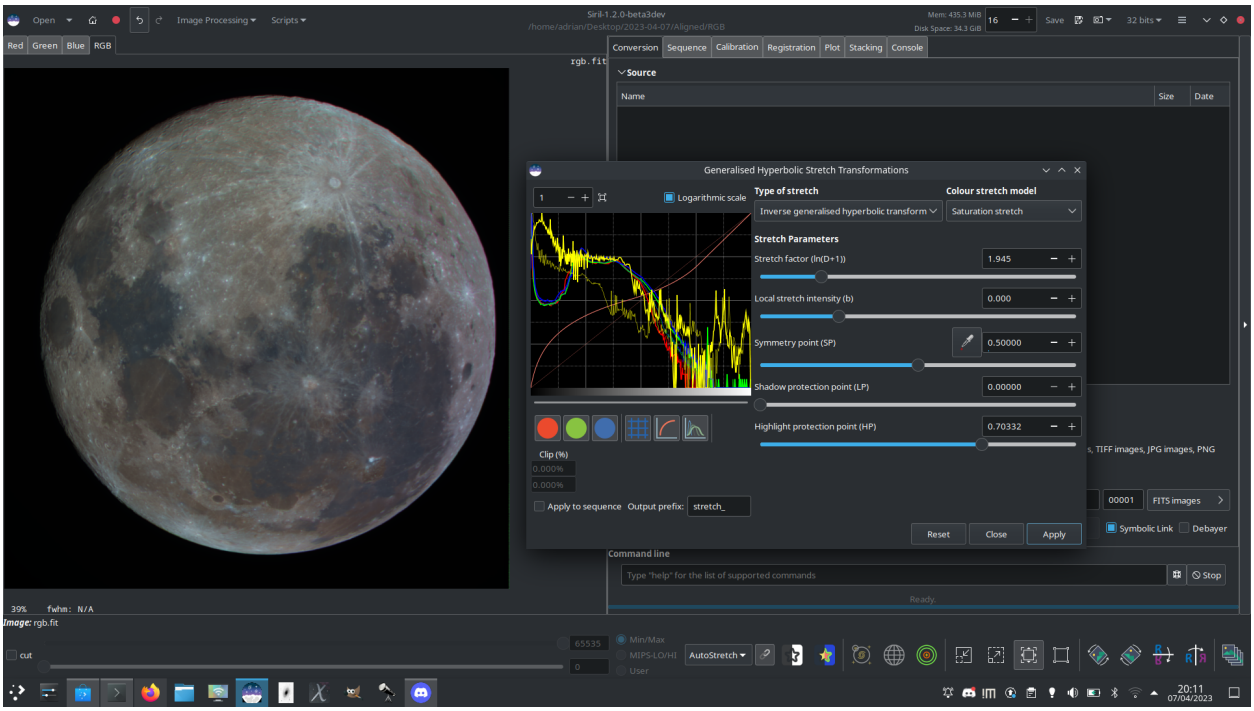


FIG. 6 – L'image ci-dessus montre comment l'application de l'outil GHS au canal de saturation permet de renforcer fortement la saturation d'une image à faible saturation tout en conservant le contrôle de la partie supérieure de l'histogramme de saturation. Cette image a été utilisée pour créer une image "Lune minérale" mettant en évidence les différentes compositions minérales des différentes régions de la surface lunaire.

Ligne de commande Siril

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

Étirement hyperbolique généralisé basé sur le travail de l'équipe ghsastro.co.uk.

L'argument **-D=** définit la force de l'étirement, entre 0 et 10. C'est le seul argument obligatoire. Les arguments facultatifs suivants permettent de mieux adapter l'étirement :

B définit l'intensité de l'étirement près du point focal, entre -5 et 15 ;

LP définit une plage de préservation des ombres entre 0 et SP où l'étirement sera linéaire, préservant les détails d'ombre ;

SP définit le point de symétrie de l'étirement, entre 0 et 1, qui est le point auquel l'étirement est le plus intense ;

HP définit une région entre HP et 1 où l'étirement est linéaire, préservant les détails de la mise en évidence et évitant le gonflement des étoiles.

Si elle est omise, B, LP et SP ont une valeur par défaut de 0.0 et HP a une valeur par défaut de 1.0.

Un argument facultatif (soit **-human**, **-even** ou **-independent**, peut être passé pour sélectionner la luminance pondérée humaine ou pondérée paire ou les canaux de couleur indépendants pour les étirements de couleur). L'argument est ignoré pour les images mono. Alternativement, l'argument **-sat** spécifie que l'étirement est effectué sur la saturation de l'image - l'image doit être en couleur et tous les canaux doivent être sélectionnés pour que cela fonctionne.

En option, le paramètre **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, V, B, RV, RB ou VB. La valeur par défaut est « tous les canaux ». Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

Ligne de commande Siril

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

Inverse un étirement hyperbolique généralisé. Il s'agit de la transformation inverse de l'étirement hyperbolique généralisé. S'il est fourni avec les mêmes paramètres, il annule une commande d'étirement hyperbolique généralisé, ce qui permet de revenir à une image linéaire. Il peut également fonctionner de la même manière que le GHT, mais pour des images en négatif

Liens : *ght*

Ligne de commande Siril

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

Étirement arcsinh modifié basé sur le travail de l'équipe ghsastro.co.uk.

L'argument **-D=** définit la force de l'étirement, entre 0 et 10. C'est le seul argument obligatoire. Les arguments facultatifs suivants permettent de mieux adapter l'étirement :

LP définit une plage de préservation des ombres entre 0 et SP où l'étirement sera linéaire, préservant les détails d'ombre ;

SP définit le point de symétrie de l'étirement, entre 0 et 1, qui est le point auquel l'étirement est le plus intense ;

HP définit une région entre HP et 1 où l'étirement est linéaire, préservant les détails de la mise en évidence et évitant le gonflement des étoiles.

Si elle est omise, LP et SP ont une valeur par défaut de 0.0 et HP a une valeur par défaut de 1.0.

Un argument facultatif (soit **-human**, **-even** ou **-independent**, peut être passé pour sélectionner la luminance pondérée humaine ou pondérée paire ou les canaux de couleur indépendants pour les étirements de couleur). L'argument est ignoré pour les images mono. Alternativement, l'argument **-sat** spécifie que l'étirement est effectué sur la saturation de l'image - l'image doit être en couleur et tous les canaux doivent être sélectionnés pour que cela fonctionne.

En option, le paramètre **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, V, B, RV, RB ou VB. La valeur par défaut est « tous les canaux ». Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

Ligne de commande Siril

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

Inverse un étirement arcsinh modifié. Il fournit la transformation inverse de MODASINH et, s'il est fourni avec les mêmes paramètres, il annule une commande MODASINH, ce qui permet de revenir à une image linéaire. Il peut également fonctionner de la même manière que MODASINH, mais pour des images en négatif

Liens : *modasinh*

Ligne de commande Siril

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

Étire linéairement l'image jusqu'à un nouveau point noir BP.
Le paramètre optionnel **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, G, B, RG, RB ou GB. La valeur par défaut est tous les canaux.
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. The clip mode can be set using the argument **-clipmode=** : values **clip**, **rescale**, **rgbblend** or **globalrescale** are accepted and the default is rgbblend

Appliquer la transformation à la séquence

Cette transformation peut facilement être appliquée à une séquence. Il suffit de définir la transformation sur l'image chargée (avec une séquence déjà chargée), puis de cocher le bouton *Appliquer à la séquence* et de définir le préfixe de sortie de la nouvelle séquence (`stretch_` par défaut). Toutes les commandes ont également une version de traitement en séquence. Chaque commande d'étirement de séquence commence par *seq* et le premier argument doit être le nom de la séquence, mais ils sinon elles sont les mêmes.

Ligne de commande Siril

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

Même commande que GHT, mais la séquence doit être spécifiée comme premier argument. En outre, l'argument optionnel **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *ght*

Ligne de commande Siril

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

Identique à la commande INVGHT, mais s'applique à une séquence qui doit être spécifiée comme premier argument. En outre, l'argument facultatif **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *invght*

Ligne de commande Siril

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

Même commande que MODASINH, mais la séquence doit être spécifiée comme premier argument. En outre, l'argument optionnel **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *modasinh*

Ligne de commande Siril

```
seqinvmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Identique à la commande INVMODASINH, mais s'applique à une séquence qui doit être spécifiée comme premier argument. En outre, l'argument facultatif **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *invmodasinh*

Ligne de commande Siril

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

Identique à la commande INVGHT, mais s'applique à une séquence qui doit être spécifiée comme premier argument. En outre, l'argument facultatif **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *linstretch*

10.1.4 Transformation par courbes

Curves Transformation is a highly versatile tool used to adjust the contrast and brightness of an image by modifying the pixel values according to a custom-defined curve. This allows for precise control over the image's stretch.

The curve is defined by a series of points, each of which can be moved to adjust the curve. The curve is interpolated between these points, and the pixel values are transformed based on it. This allows for a wide range of transformations to be applied to the image, from simple linear stretches to complex non-linear adjustments.

There are two interpolation algorithms available in the Curves Transformation dialog : linear and cubic spline.

FIG. 7 – Dialog box of the Curves Transformation

Théorie

— Linear interpolation is a simple interpolation that connects the points with straight lines. For each pair of points, the slope of the line connecting the points is calculated :

$$\text{slope} = \frac{y_2 - y_1}{x_2 - x_1}. \tag{10.3}$$

The pixel values are then transformed by evaluating the line at the original pixel value :

$$\text{pixel} = \text{slope} \times (\text{original} - x_1) + y_1. \tag{10.4}$$

— Cubic spline curves are more complex curves that are defined by a series of control points. For pixel value x between two control points x_i and x_{i+1}, the curve is defined by the following equation :

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{10.5}$$

For $x_i < x < x_{i+1}$, the coefficients a_i, b_i, c_i, and d_i are calculated by solving a system of equations derived from the conditions of continuity and smoothness at each internal point. These conditions are :
— The spline must be continuous at each internal point,
— The first derivative of the spline must be continuous at each internal point,
— The second derivative of the spline must be continuous at each internal point
— Since Curves Transformation uses natural cubic splines, the second derivative at both endpoints is 0.

Avertissement

Curves Transformation is only available as a GUI tool and cannot be used from the command line.

Curves Transformationcan display the histogram of the image in two modes : linear and logarithmic. You can swap between these modes by clicking the *Logarithmic scale* checkbox at the top of the histogram. Logarithmic scale is useful for images with a wide dynamic range, as it allows you to see the histogram in the shadows and highlights more clearly.

Astuce

If a ROI is set, the histogram preview will not update to show the impact of the stretch on the ROI. This is because that behaviour could be misleading : if the ROI is not typical of the image overall, adjusting the ROI histogram to a suitable level would result in a badly adjusted histogram for the overall image and potentially a burned-out or excessively dark look to the result. When in ROI mode the stretch parameters should be adjusted by eye. If it is desired to check the histogram for the stretch as applied to the image as a whole, the ROI should be cleared.

Voici quelques utilisations courantes de l'outil Transformation par courbes :

— **Courbe en "S" :** Cette courbe sert à augmenter le contraste d'une image. En accentuant la pente de la courbe au milieu de l'histogramme, le contraste est augmenté.

— **BP & WP adjustments :** The black point (BP) and white point (WP) of an image can be adjusted by moving the first and last control points of the curve. This allows for the shadows and highlights of the image to be adjusted independently.

FIG. 8 – Un exemple d'une courbe en "S"

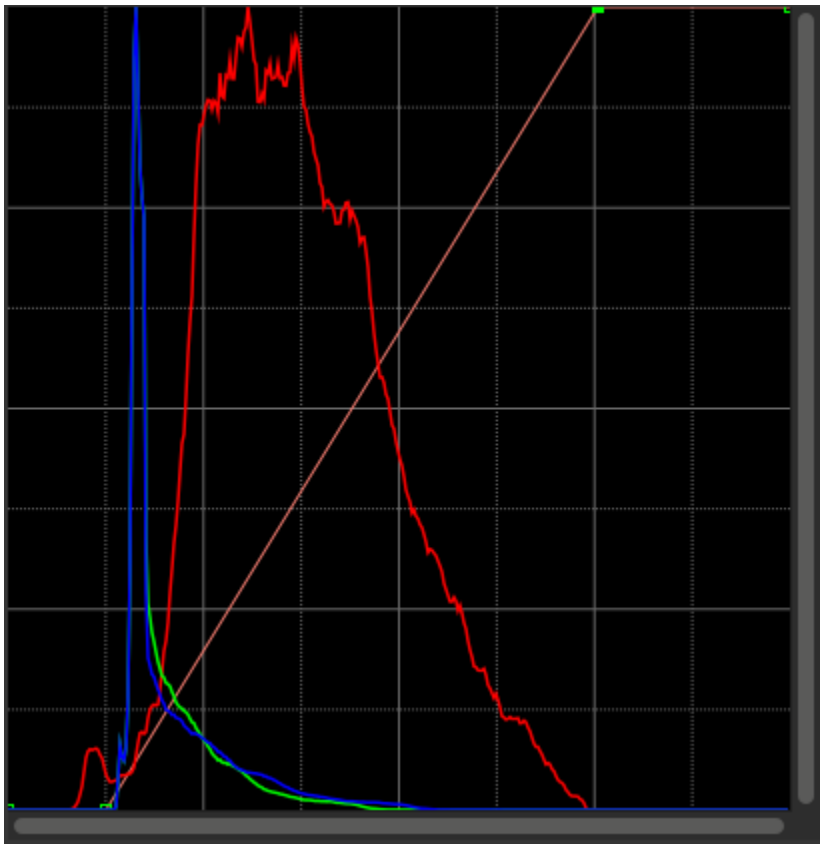


FIG. 9 – Exemple de réglage du point noir et du point blanc d'une image

— **Targeted adjustments :** By adding control points at specific locations in the histogram, targeted adjustments can be made to the image. For example, the shadows can be darkened without affecting other parts of the image.

Fig. 10 – An example of a targeted adjustment to an image

Astuce

The Curves Transformation tool is best used on an image that has already been stretched to some extent. This allows for more precise control over the image's contrast and brightness.

This transformation can easily be applied to a sequence. All you have to do is define the transformation on the loaded image while a sequence is loaded and check the *Apply to sequence* button. Similarly to other stretching tools, you can define the output prefix of the new sequence (`curve_` by default).

10.2 Étalonnage des couleurs

Siril propose trois façons de récupérer les couleurs de votre image :

— *Étalonnage manuel*
— *Étalonnage photométrique des couleurs*, également appelé PCC.
— *Étalonnage des couleurs par spectrophotométrie*, également appelé SPCC. Il s'agit de la méthode la plus précise et celle recommandée

Ici, "récupérer" signifie rééquilibrer les canaux RVB pour se rapprocher le plus possible des vraies couleurs de l'objet photographié.

Astuce

PCC et SPCC sont deux méthodes qui nécessitent une*image résolue par astrométrie*. Il n'est pas possible d'accéder à ces outils sans ce prérequis.

10.2.1 Étalonnage manuel des couleurs

Avertissement

L'étalonnage des couleurs **doit** être effectué sur une image linéaire dont l'histogramme n'a pas encore été étiré. Dans le cas contraire, les couleurs obtenues ne sont pas garanties comme étant correctes.

La méthode manuelle utilise la fenêtre suivante :

La première étape concerne le **fond** de votre image. L'objectif est d'égaliser les couches RVB afin que l'arrière-plan apparaisse avec une couleur grise neutre.

Après avoir effectué une sélection dans votre image (dans une zone peu étoilée), la zone est prise en compte en cliquant sur le bouton *Utiliser la sélection actuelle*. Les coordonnées du rectangle sont affichées. Ensuite, *Neutralisation de fond de ciel* va calculer la médiane de chaque canal et les égaliser.

La deuxième étape concerne les **objets lumineux** de l'image. Vous pouvez à nouveau modifier l'histogramme de deux manières :

- Manuellement, avec **Référence des blancs** et les 3 coefficients R, G et B, selon votre propre goût.
- Automatiquement, en sélectionnant une zone rectangulaire avec des objets contrastés (de la même manière que précédemment)

Deux curseurs vous permettent de modifier la limite de rejet des pixels trop sombres et trop clairs de la sélection.

Comme il s'agit d'un processus d'essai et d'erreur, vous pouvez annuler le résultat avec le bouton *Annuler* (en haut à gauche) et ensuite essayer avec d'autres sélections ou coefficients jusqu'à ce que vous soyez satisfait.

10.2.2 Étalonnage des couleurs par photométrie

Avertissement

L'étalonnage des couleurs par photométrie **doit** impérativement être effectué sur une image linéaire dont l'histogramme n'a pas encore été étiré. Sinon, les mesures photométriques seront fausses et les couleurs obtenues sans garantie d'être correctes.

Une autre façon de récupérer les couleurs est de comparer la couleur des étoiles dans l'image avec leur couleur dans les catalogues, pour obtenir la couleur la plus naturelle de façon automatique et non subjective. C'est l'outil PCC (Photometric Color Calibration) (`Ctrl` + `Shift` + `P`). Il ne peut fonctionner que pour des images prises avec un ensemble de filtres rouges, verts et bleus pour les couleurs, ou avec la couleur du capteur. Pour identifier les étoiles de l'image avec celles du catalogue, une solution astrométrique est nécessaire. Veuillez consulter la *documentation du module de résolution astrométrique*.

Cette méthode est moins précise que l'étalonnage des couleurs par spectrophotométrie expliqué ci-dessous, mais elle peut être réalisée à l'aide de catalogues locaux et constitue donc la meilleure option lorsqu'une connexion Internet n'est pas disponible.

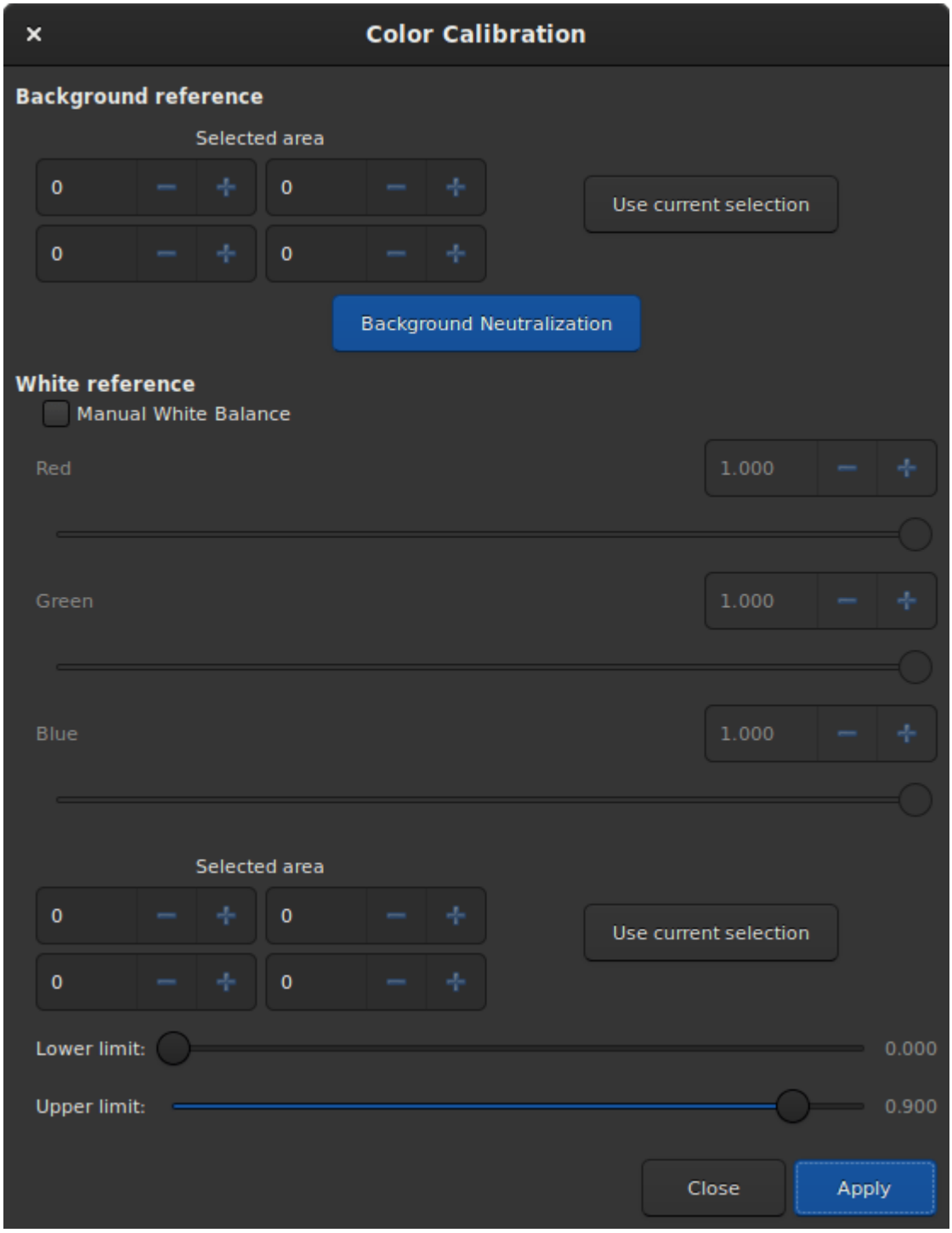


FIG. 11 – Fenêtre de dialogue d'étalonnage manuel des couleurs.

> **Note**
>
> Cette technique dépend fortement du type de filtre utilisé. L'utilisation de différents types de filtres R, V, B ne fera pas une grande différence, mais l'utilisation d'un filtre contre la pollution lumineuse ou l'absence de filtres bloquant les IR fera dévier la solution de manière significative et ne donnera pas les couleurs attendues.

Depuis la version 1.4, les deux outils fonctionnent indépendamment : il est possible d'exécuter l'analyse photométrique et la correction des couleurs de l'image seulement si l'image a déjà été résolue astrométriquement. Cela signifie également que des catalogues différents peuvent être utilisés pour PCC et l'astrométrie. L'outil est aussi disponible sous la forme de la commande `pcc`, ce qui permet de l'intégrer dans des scripts de post-traitement d'images.

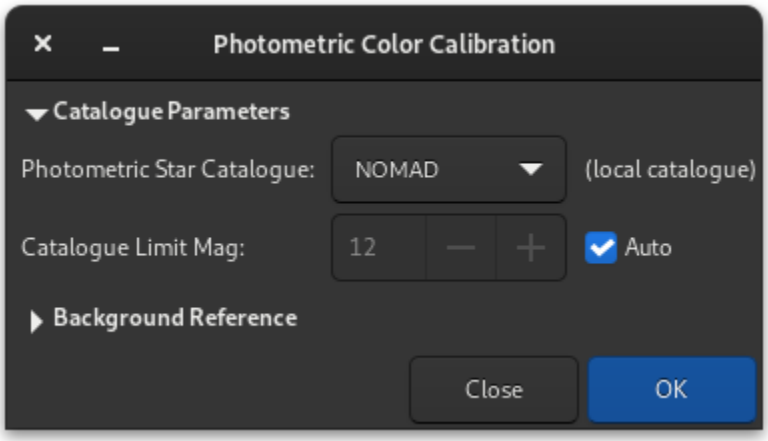


FIG. 12 – Fenêtre de dialogue Étalonnage des couleurs par photométrie.

Si l'image a été précédemment résolue astrométriquement, activez la fonction *annotations* pour vérifier que les catalogues sont alignés avec l'image. Si la solution astrométrique n'est pas assez bonne, veuillez refaire une astrométrie.

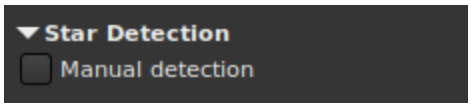
— La section **Paramètres du catalogue** vous permet de choisir le catalogue photométrique à utiliser, NOMAD, APASS ou Gaia DR3, ainsi que la magnitude limite.

> **Astuce**
>
> The NOMAD and GAIA catalogs can be *locally installed*, while the APASS catalog requires internet access to obtain its contents.
> Le catalogue Gaia local utilisé pour PCC est le catalogue astrométrique, qui est un téléchargement beaucoup plus petit que le catalogue SPCC complet. S'il est disponible, il sera affiché par défaut.

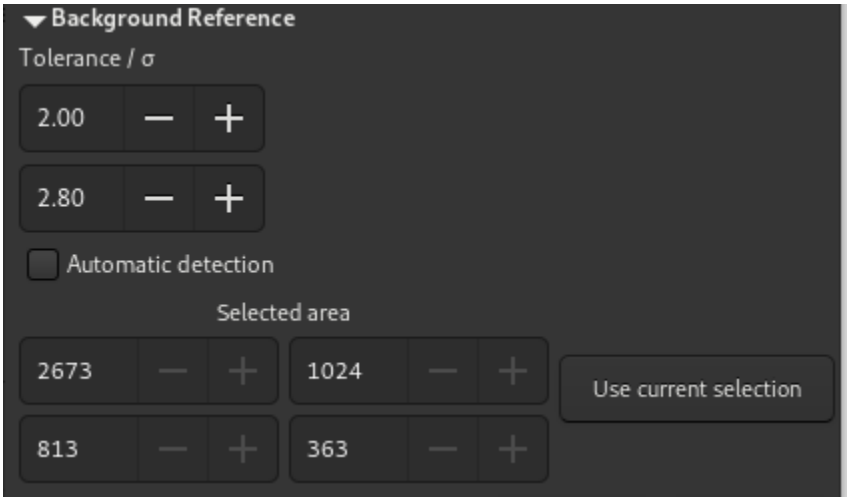
> **Note**
>
> Whereas the NOMAD and APASS catalogs use the catalogued B-V index and estimate the effective temperature from this value, Gaia provides an accurately modelled effective temperature therefore when this catalog is used the Teff field is used directly, giving more accurate results.

— La section **Détection d'étoiles** vous permet de sélectionner manuellement les étoiles qui seront utilisées pour l'analyse photométrique. Il est préférable d'en avoir des centaines au moins, la sélection individuelle n'est donc pas idéale.

— Si vous le souhaitez, la **référence de fond de ciel** peut être sélectionnée manuellement comme décrit dans*Étalonnage manuel des couleurs*. Cela peut être utile dans le cas d'images de nébuleuses où les parties



du ciel en arrière-plan sont petites.



Lorsque suffisamment d'étoiles sont trouvées et que la solution astrométrique est correcte, le PCC affiche ce type de texte dans l'onglet Console :

```
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
```

On peut comprendre que 433 étoiles ont été sélectionnées dans le catalogue et l'image pour l'analyse photométrique, mais étrangement, seules 363 ont été réellement utilisées, 70 ayant été exclues. La ligne *Distribution des erreurs* explique la raison de leur exclusion : 18 n'ont pas été trouvées à la position attendue, 48 étaient trop grandes et 4 étaient probablement saturées. Il est très courant que de nombreuses étoiles soient rejetées car elles ne répondent pas aux exigences strictes d'une analyse photométrique valide.

Nous pouvons également voir que le PCC a trouvé trois coefficients à appliquer aux canaux de couleur pour corriger l'équilibre des blancs. La *déviation* ici, qui est l'écart absolu moyen de la correction des couleurs pour chacune des étoiles de l'ensemble photométrique, est modérément élevée. Sur des images bien calibrées, sans gradient, avec des filtres corrects et sans nébuleuse de couleur recouvrant toute l'image, la déviation se rapprocherait de 0.

Ligne de commande Siril

```
pcc [-limitmag=[+-]] [-catalog=] [-bgtol=lower,upper]
```

Exécute la Correction de Couleur par Photométrie sur l'image chargée et résolue astrométriquement.

La magnitude limite des étoiles est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement une valeur positive absolue pour la magnitude limite.
The star catalog used is NOMAD by default, it can be changed by providing **-catalog=apass**, **-catalog=localgaia** or **-catalog=gaia**. If installed locally, the remote NOMAD (the complete version) can be forced by providing **-catalog=nomad**
La tolérance aux valeurs aberrantes de la référence de fond de ciel peut être spécifiée en unités sigma en utilisant **-bgtol=lower,upper** : ces valeurs sont par défaut de -2,8 et +2,0

10.2.3 Étalonnage des couleurs par spectrophotométrie

Avertissement

L'étalonnage des couleurs par photométrie **doit** impérativement être effectué sur une image linéaire dont l'histogramme n'a pas encore été étiré. Sinon, les mesures photométriques seront fausses et les couleurs obtenues sans garantie d'être correctes.

L'étalonnage des couleurs par spectrophotométrie (`Ctrl` + `Shift` + `C`) est la méthode d'étalonnage des couleurs la plus récente disponible dans Siril. Cette méthode utilise les nombreuses données spectrales disponibles dans le catalogue en ligne Gaia DR3 [GaiaDR3]. Il est possible d'y accéder soit en interrogeant directement le catalogue en ligne, soit en téléchargeant un extrait local et en interrogeant le catalogue local.

Avertissement

Notez que lorsque les archives Gaia sont hors ligne pour maintenance ou en raison d'une panne, la fonctionnalité SPCC de Siril ne sera pas disponible via le catalogue en ligne. Heureusement, les archives sont normalement très fiables, mais un indicateur d'état est intégré dans la boîte de dialogue SPCC. L'état des archives est vérifié au démarrage de la boîte de dialogue et peut être revérifié en cliquant sur le bouton d'état.

Le catalogue Gaia SPCC hors ligne fonctionnera toujours si l'archive Gaia est hors ligne.

Astuce

Quelle est la différence entre SPCC et PCC ? Quand devrais-je utiliser l'un plutôt que l'autre ? SPCC est une version plus précise de PCC et rend ce dernier obsolète. SPCC prend en compte le capteur et les filtres de votre matériel. Par conséquent, la couleur produite est plus proche de la "réalité". L'exemple de l'image ci-dessous illustre la différence de résultats.

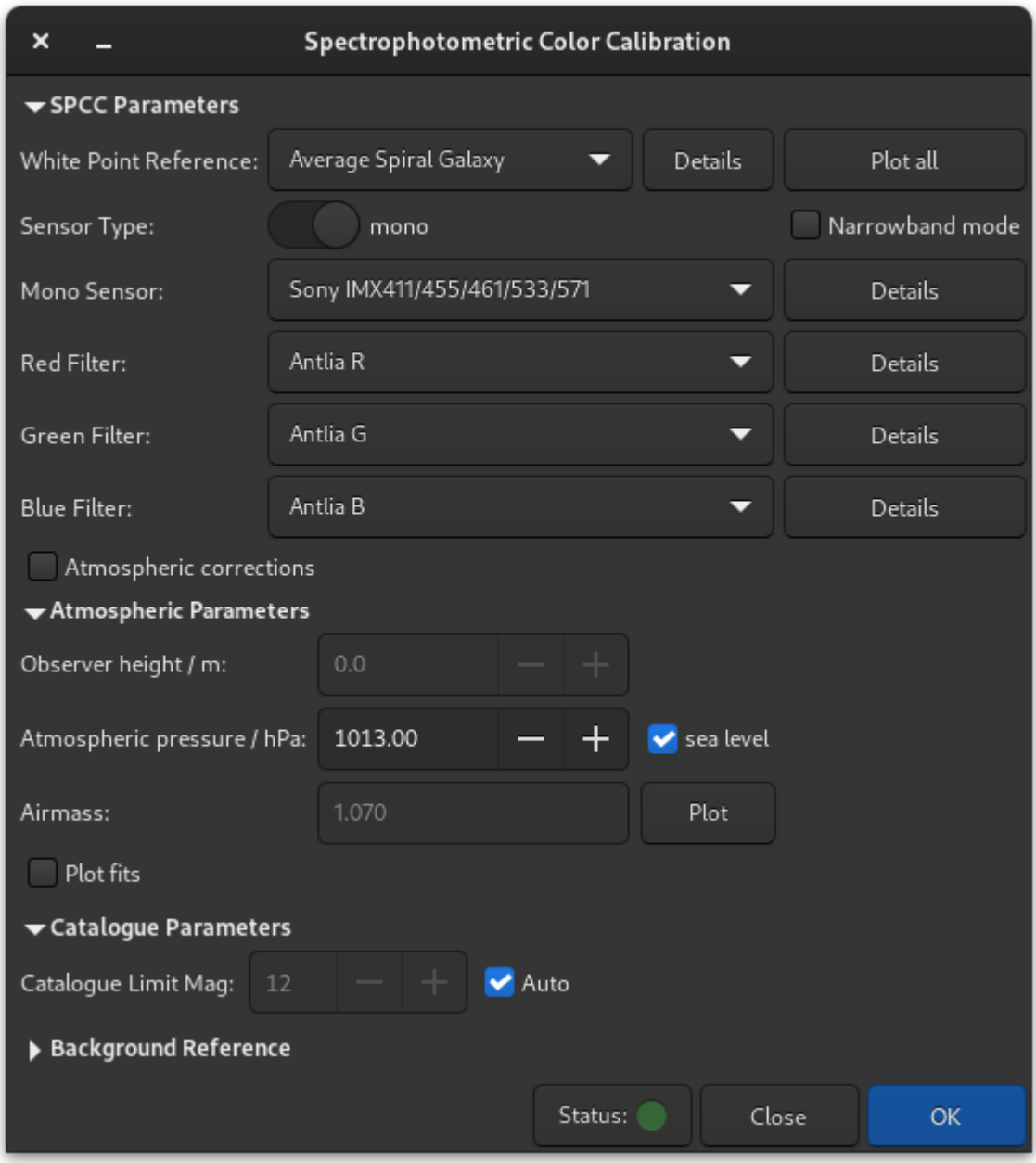


FIG. 13 – Fenêtre de dialogue Étalonnage des couleurs par spectrophotométrie.

FIG. 14 – Comparaison entre PCC (à gauche) et SPCC (à droite) : cliquer pour agrandir. (Avec l'aimable autorisation de Ian Cass)

Catalogue local SPCC

À partir de la version 1.4.0, un catalogue SPCC hors ligne est disponible utilisant les données de Gaia DR3. Notez que le catalogue est divisé en 48 fichiers couvrant chaque HEALpixel de niveau 1.

Théorie

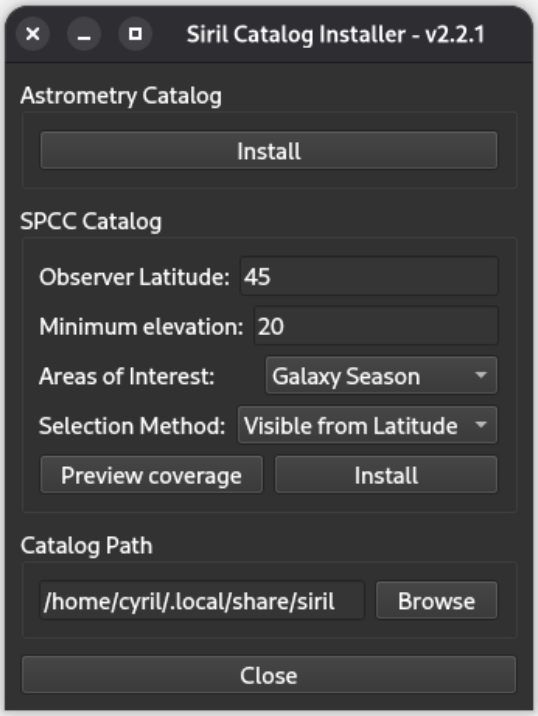
HEALpix (Hierarchical Equal Area isoLatitude Pixelisation) is an algorithm for pixelising a sphere based on subdivision of a distorted rhombic dodecahedron. Mathematical details can be found on Wikipedia [Wiki_HEALPIX]. Gaia sources use a Level 12 NESTED HEALpix scheme and the HEALpixel number is encoded into the source_id. The specification of the Gaia DR3 catalogue extracts and their file format is documented here (PDF).

La nature imbriquée du schéma signifie que les HEALpixels qui sont proches les uns des autres dans le ciel ont des numéros qui sont également proches. La propriété hiérarchique signifie également qu'il est possible d'indexer les sources dans les HEALpixels à un niveau HEALpixel profond et de diviser le catalogue en morceaux à un niveau moins profond, tout en continuant à prendre en charge un algorithme de recherche de catalogue hautement efficace.

It is possible to download the entire catalog or only the chunks you need. The folder location to store the catalog files is set in *Preferences->Astrometry*.

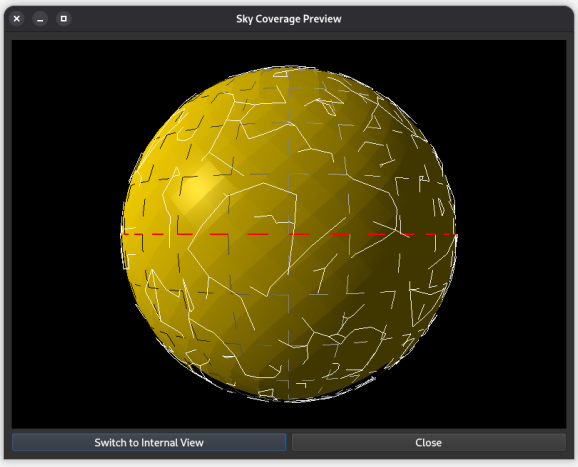
Scripts Sirilpy

La façon la plus simple d'installer le catalogue est d'utiliser le script Python intégré `Siril_Catalog_Installer.py` dans le menu *Scripts->Scripts Python*.



Cela fournit une interface qui vous permet d'installer soit le catalogue entier, soit uniquement les morceaux qui sont visibles depuis votre latitude d'observation au-dessus d'une certaine élévation, soit uniquement des ensembles de morceaux correspondant à certains thèmes (*Voie Lactée*, *Triangle d'Été*, *Saison des Galaxies* etc.) Sélectionnez la latitude / élévation ou la zone d'intérêt si vous le souhaitez, puis sélectionnez la méthode de sélection (*Tout*, *Visible depuis la Latitude* ou *Zone d'Intérêt*).

Vous pouvez prévisualiser la couverture en utilisant le bouton *Prévisualiser la couverture*.



Enfin, en cliquant sur *Installer*, vous téléchargerez, vérifierez, décompresserez et installerez les morceaux sélectionnés et paramétrerez également le chemin du catalogue dans les préférences de Siril. Un chemin de catalogue

par défaut est suggéré dans le widget de saisie de texte, mais peut être modifié pour un emplacement différent si vous préférez.

Si vous souhaitez installer manuellement les fichiers du catalogue SPCC hors ligne, ils peuvent être téléchargés depuis Zenodo. Il est possible de télécharger soit des HEALpixels individuels de niveau 1, soit l'ensemble du catalogue sous forme d'archive.

Astuce

Lorsque vous téléchargez "Tous les fichiers" à partir de l'enregistrement Zenodo, le téléchargement est une archive zip que vous devrez extraire, cependant l'archive zip est juste un moyen pratique de regrouper tous les fichiers individuels ; les fichiers de données à l'intérieur de l'archive zip sont eux-mêmes compressés avec la compression bzip2, et vous devrez décompresser les fichiers .bz2 individuels avant que Siril ne puisse les utiliser. La prise en charge de ce format de compression est disponible par défaut sous Linux et MacOS, et est assurée sous Windows par divers programmes d'archivage, dont 7-Zip et Pea-Zip, qui sont tous deux des logiciels libres et open source.

Tous les fichiers compressés sont accompagnés de sha256sums et il y a un fichier contenant tous les sha256sums des fichiers non compressés, pour une validation supplémentaire. L'enregistrement Zenodo fournit également une référence DOI qui peut être utilisée pour citer l'ensemble de données si vous l'utilisez dans un travail universitaire.

Siril utilise un extrait optimisé du produit datalink xp_sampled de Gaia DR3. Comme pour l'extrait astrométrique, le catalogue hors ligne est limité aux 127 sources les plus brillantes par HEALpixel de niveau 8. Le catalogue contient moins de sources que l'extrait astrométrique car les spectres xp_sampled ne sont généralement fournis que pour les sources plus brillantes que la magnitude 17,6. Par conséquent, davantage de HEALpixels dans les zones moins peuplées du ciel contiennent moins de 127 sources par rapport à l'extrait astrométrique (c'est-à-dire que ces HEALpixels contiennent toutes les sources Gaia DR3 disponibles avec des données xp_sampled). Cependant, cette approche évite toujours la surpopulation du catalogue dans les parties extrêmement denses du ciel tout en fournissant le meilleur rapport signal/bruit. Dans les HEALpixels avec moins de 127 sources xp_sampled, le catalogue local est aussi complet que si l'on utilisait directement l'archive Gaia en ligne.

Le xp_sampled est converti de données float32 en float16 avec un octet supplémentaire définissant l'exposant à appliquer aux données xp_sampled pour la source, afin de surmonter les limitations des exposants exprimables en float16. Cette approche est entièrement justifiable compte tenu des marges d'erreur sur les données xp_sampled et ne fait aucune différence pratique sur la précision des résultats. Cela signifie que nous pouvons fournir un catalogue SPCC local hautement efficace et optimisé pour son usage spécifique, dans moins de 21 Go de données.

Comment cela marche

SPCC nécessite la connaissance de votre capteur et des filtres RVB que vous utilisez. Ceux-ci sont fournis par un *dépôt en ligne* que Siril synchronisera, soit automatiquement au démarrage, soit manuellement lorsque cela est nécessaire. Les informations sur les capteurs et les filtres sont mises à jour par la même méthode de synchronisation que celle utilisée pour le dépôt de scripts en ligne. (Cela signifie que lorsque des données sur de nouveaux filtres ou capteurs sont disponibles, elles peuvent être ajoutées au dépôt sans nécessiter de mise à jour de l'application.)

In the GUI you select your sensors and filters from the widgets in the SPCC dialog. Don't worry if there isn't an exact match for your equipment, just pick the closest option, or the appropriate default option. You also need to select a white reference. The default reference is the Average Spiral Galaxy reference which is suitable for a wide range of astrophotographic scenes, however there is an extensive range of galaxy and star types to choose from. The Sun's spectral type is G2(v) so if you want to balance your image using sunlight as a white reference, you would pick **Star, type G2(v)** from the list.

SPCC utilise ensuite les spectres stellaires de Gaia DR3 et la connaissance de votre capteur d'imagerie et de vos filtres pour calculer pour chaque étoile du catalogue qui correspond à une étoile détectée dans l'image par Siril, le flux attendu

dans chaque canal de couleur. Il compare ensuite ce flux avec le flux réel mesuré dans chaque canal en utilisant les capacités photométriques de Siril.

Compte tenu de la connaissance du capteur et du filtre, SPCC calcule le flux attendu dans chaque canal pour la référence des blancs spécifiée. Un ajustement linéaire robuste est obtenu pour donner le meilleur ajustement des rapports de flux R/V et B/V du catalogue à l'image pour chaque étoile et pour la référence des blancs. Cet ajustement est utilisé pour dériver des coefficients de correction qui sont appliqués de manière multiplicative à chaque canal, ce qui permet d'obtenir des canaux de couleur spectrophotométriquement exacts.

Votre image doit être résolue pour que SPCC fonctionne : si ce n'est pas le cas, vous devez le faire à l'aide de l'outil prévu à cet effet. Il est important de s'assurer que les informations relatives à la résolution astrométrique sont correctes, car certains logiciels sont connus pour ajouter des données WCS inexactes aux images.

Interface Graphique

- **Sélection du capteur** Pour sélectionner votre capteur, assurez-vous que le bouton mono / couleur est correctement réglé. Vous verrez alors le menu déroulant approprié pour choisir parmi les capteurs disponibles.
- **Sélection de filtres** SPCC peut fonctionner selon deux modes.
  - Le mode par défaut est le mode large bande. Dans ce mode, la case *mode bande étroite* doit être décochée. Vous pouvez choisir des filtres rouge, vert et bleu (pour les images composées réalisées avec un capteur mono) ou des filtres couleur, par exemple des filtres de pollution lumineuse, pour les images réalisées avec un capteur couleur.

    > **Avertissement**
    >
    > Si vous sélectionnez un capteur APN couleur (par exemple, un Canon EOS 600D), un widget supplémentaire deviendra visible pour sélectionner un filtre passe-bas. Cela vous permet de préciser si votre appareil photo a été modifié pour l'astronomie ou non. Vous **devez** sélectionner une option ici ou le processus se plaindra que vous n'avez pas défini tous les filtres nécessaires !
    > Des options existent pour les filtres passe-bas d'origine Canon et Nikon, ainsi que pour le populaire filtre de modification astronomique Baader BCF qui laisse passer le Ha et Sii mais bloque toujours les longueurs d'onde IR plus longues, et "Full spectrum" qui est modélisé comme un filtre parfaitement transparent.
    > Si vous avez un appareil photo non modifié d'un modèle ou d'une marque différente, sélectionnez n'importe lequel des filtres passe-bas Canon ou Nikon : l'effet est très mineur car ces longueurs d'onde sont de toute façon à la limite de la perception visuelle humaine.

  - En cochant la case *Mode bande étroite*, vous activez le mode bande étroite. Celui-ci est destiné soit aux images composées à partir de filtres à bande étroite utilisés avec un capteur mono, soit aux images réalisées à l'aide d'un capteur couleur avec un filtre à bande étroite duo, tri-bande ou quadri-bande. Dans ce mode, les commandes disponibles changent et, pour chaque canal de couleur, vous entrez la longueur d'onde nominale et la largeur de bande de la bande passante du filtre. Pour les filtres mono bande ultra étroits, la bande passante peut être de 3 nm seulement ; pour un filtre OSC quadribande comme l'Altair QuadBand V2, les bandes passantes peuvent atteindre 35 nm. Notez que pour une composition HOO où deux canaux sont réglés sur les mêmes données, la longueur d'onde nominale et la largeur de bande doivent également être égales dans l'interface de SPCC.

    > **Astuce**
    >
    > Certains fabricants spécifient une longueur d'onde centrale et une FWHM. Il n'y a pas de problème à utiliser la FWHM comme largeur de bande : ces filtres ont des coupures très nettes.

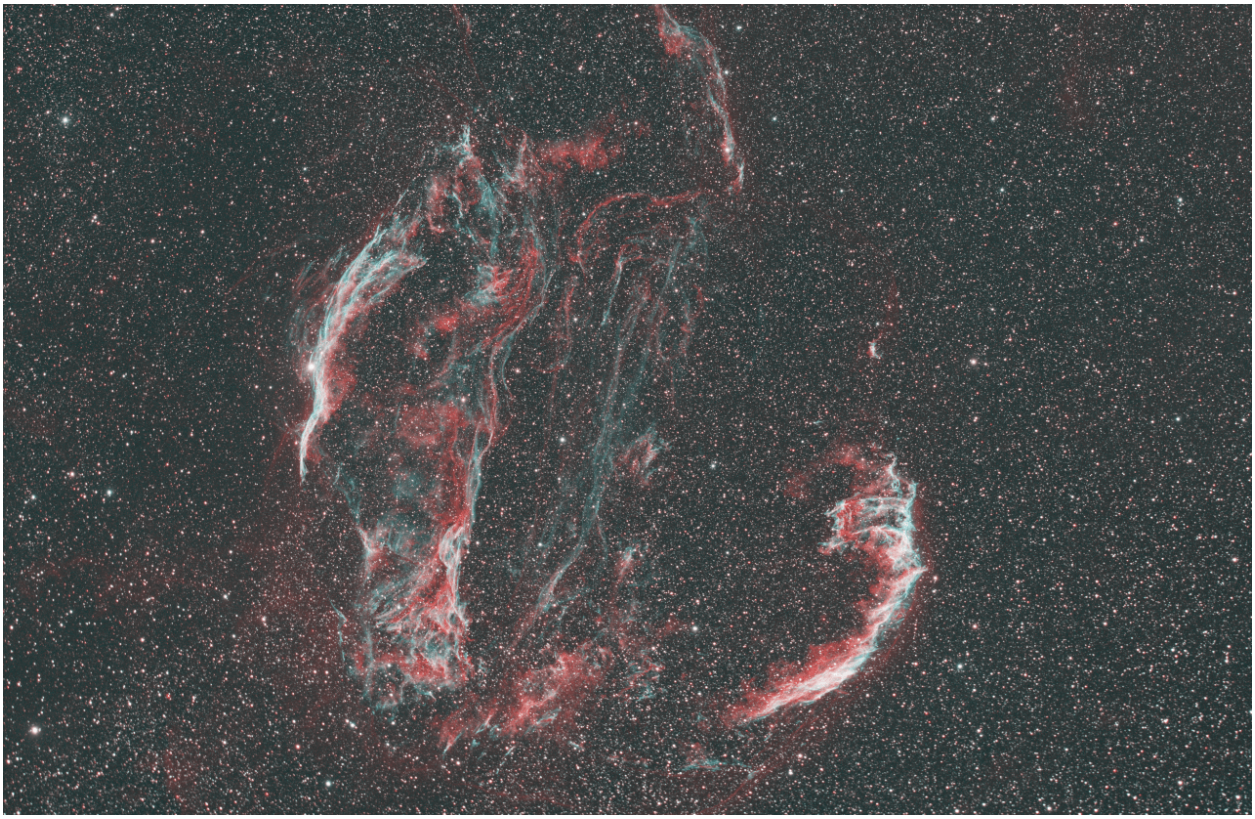


FIG. 15 – Image HOO calibrée. Image de Cyril Richard.

> **Avertissement**
>
> Ne vous attendez pas à récupérer la palette Hubble pour l'imagerie SHO en utilisant les longueurs d'onde des filtres SII,Hα et OIII respectivement. Le résultat sera une image avec une énorme dominante verte. Cela s'explique facilement par le fait que la raie d'émission SII est beaucoup plus faible que celle de l'hydrogène, et que SPCC donne une représentation des intensités réelles. Mais ce n'est pas le cas dans la palette Hubble. En fait,*l'étalonnage manuel des couleurs* donnera de meilleurs résultats.

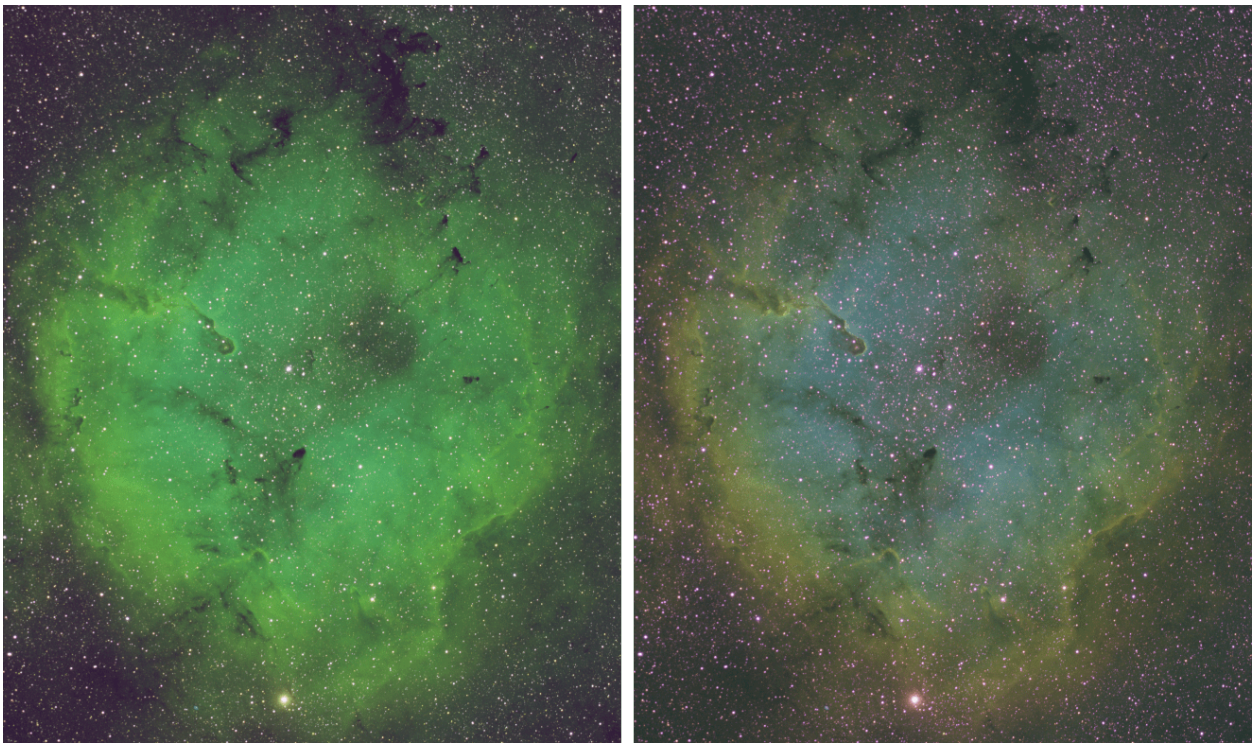


Fig. 16 – Image SHO étalonnée par SPCC comparée à la même image étalonnée manuellement. La nébuleuse entière a été prise comme référence des blancs lors de l'étalonnage manuel. Image de Cyril Richard.

— **Sélection du filtre passe-bas (LPF)** Les reflex numériques contiennent un filtre passe-bas (parfois aussi appelé "miroir chaud"). Ces filtres réduisent la transmittance aux longueurs d'onde qui intéressent les astronomes (Ha à 656nm et S-II à 674nm). Si le capteur couleur sélectionné est un APN, un menu déroulant vous permet de sélectionner le profil LPF approprié. Il existe des options pour les filtres LPF standard ainsi que pour les filtres LPF astro-modifiés et un modèle de filtre **Spectre complet** idéal pour les cas où le filtre LPF a été complètement supprimé.

— **Selection of White Reference** SPCC requires an absolute white reference spectrum. The default is **Average Spiral Galaxy** and the source spectra used to create this white reference are taken from the SWIRE templates [SWIRE] in a manner consistent with other astrophotography software providing the same white reference. A wide range of other white references is available, covering the full range of galaxy and star classifications [Stellar]. If you wish to use sunlight as your white reference, you would choose the white reference **Star, type G2(v)** as the Sun is a type G2(v) star.

> **Astuce**
>
> **Résumé des classifications spectrales stellaires** Les classifications stellaires se composent de deux parties, un type Morgan-Keenan et un indice de luminosité.

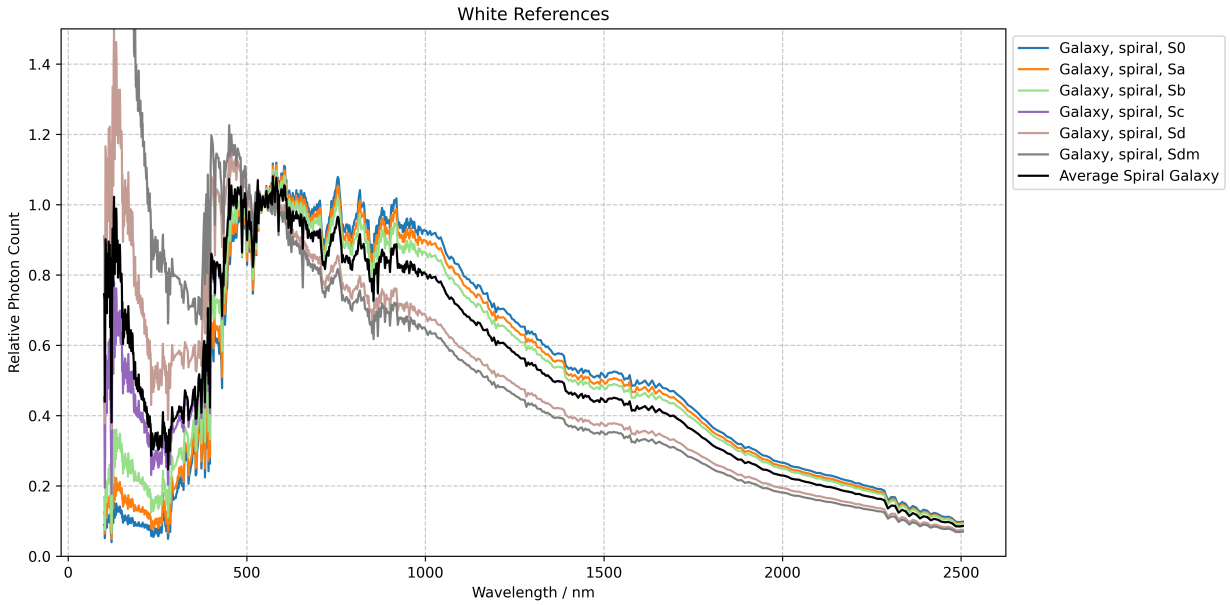


FIG. 17 – Graphiques montrant les données de la référence de blanc des galaxies spirales. À environ 350 nm, les données des galaxies spirales moyennes deviennent identiques à celles des galaxies Sc, qui sont également une bonne représentation de la référence de blanc.

La première partie de la classification spectrale (G2 dans le cas du Soleil) prend l'une des lettres suivantes : O, B, A, F, G, K, M. O représente les étoiles bleues extrêmement chaudes, tandis que M représente les étoiles rouges froides : Le soleil se situe à peu près au milieu du spectre. Le nombre représente des cas intermédiaires, par exemple une étoile B5 est à mi-chemin entre le type B et le type A.
La deuxième partie de la classification spectrale est l'indice de luminosité, qui va de i à v. Les étoiles ayant un indice de luminosité i sont des supergéantes, tandis que les étoiles ayant un indice de luminosité v sont des naines. Les étoiles de la séquence principale, comme le soleil, ont un indice de luminosité iv.

— **Correction atmosphérique** La SPCC de Siril prend en charge la correction atmosphérique. Lorsque nous prenons des images depuis la Terre, nous les prenons à travers l'atmosphère. Celle-ci n'a pas une transmittance parfaite et agit donc comme un autre filtre, non optionnel, dans la chaîne d'imagerie entre le capteur et l'objet astronomique. La correction de cette atmosphère est un choix artistique que chaque astrophotographe doit faire, mais l'option est fournie.

Théorie

L'extinction atmosphérique a plusieurs origines. Les plus importantes sont les suivantes :
- Diffusion de Rayleigh. Il s'agit de la diffusion élastique de la lumière par des particules qui sont petites par rapport à la longueur d'onde de la lumière. La forte dépendance de la diffusion de Rayleigh par rapport à la longueur d'onde ($\approx^{-4}$) signifie que les courtes longueurs d'onde (bleues) sont diffusées plus fortement que les grandes longueurs d'onde (rouges).
- Diffusion des aérosols. Il s'agit de la diffusion de la lumière par des particules plus grandes que la longueur d'onde de la lumière. Ce phénomène est très variable mais (en l'absence d'effets significatifs à court terme de la poussière ou de la fumée) relativement plat sur le plan spectral et moins important que la diffusion de Rayleigh.
- Raies d'absorption moléculaire.

Siril ne modélise que la diffusion de Rayleigh. Il s'agit de la contribution la plus importante dans la plupart des conditions atmosphériques, et elle est hautement prévisible, ce qui la rend facile à modéliser sans exiger

FIG. 18 – NGC 4414 est un excellent exemple de galaxie de type Sc, le type le plus proche de la galaxie spirale moyenne utilisée comme référence de blanc par défaut. Crédit image : NASA, ESA, W. Freedman (U. Chicago) et al, & the Hubble Heritage Team (AURA/STScI), SDSS ; Processing : Judy Schmidt.

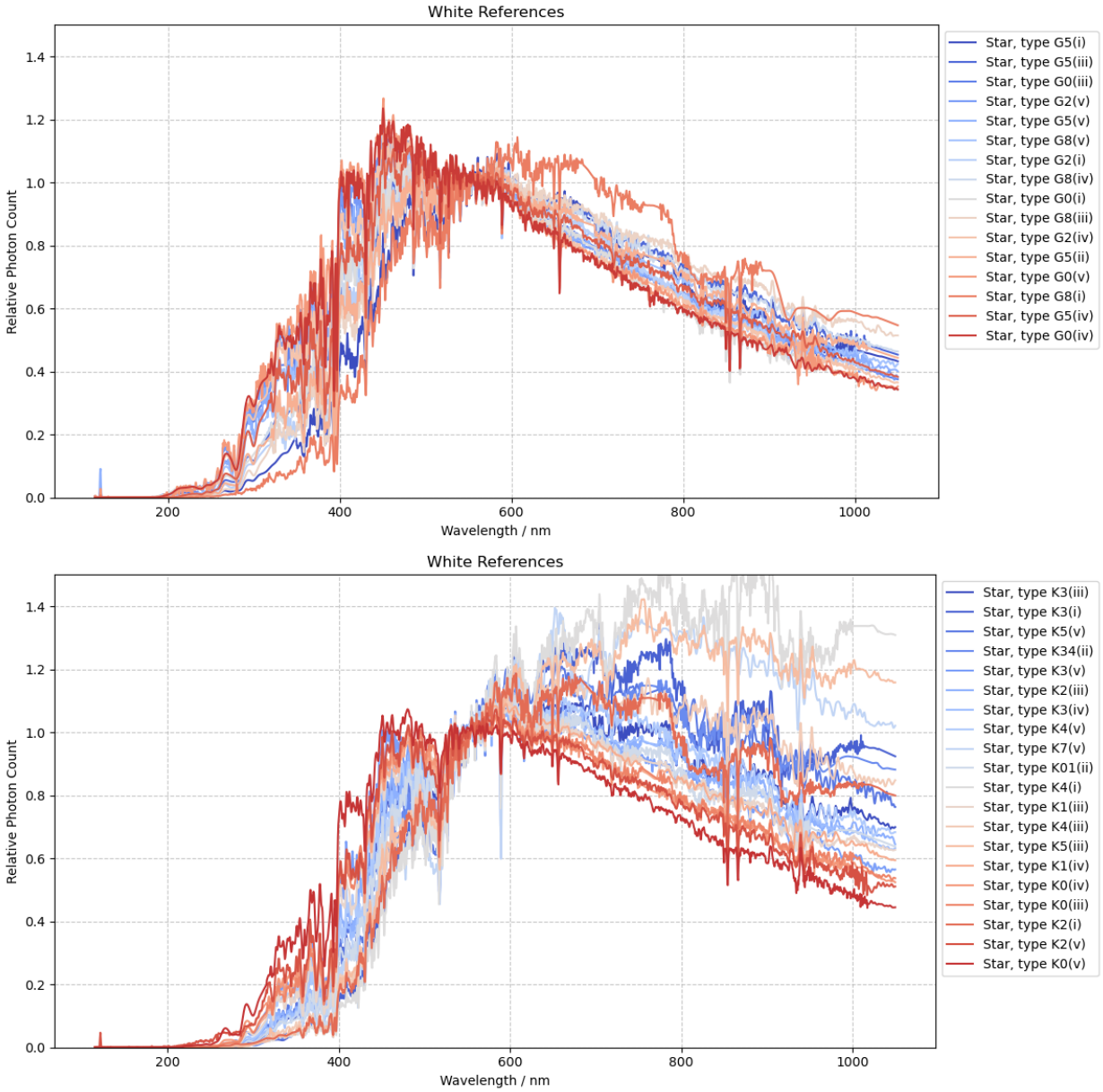


FIG. 19 – Graphiques montrant les données de référence de blanc pour un ensemble de deux classes d'étoiles différentes, G et K.

FIG. 20 – Différence d'étalonnage des couleurs en fonction du choix de la référence de blanc. À gauche, une étoile de type M, à droite la galaxie spirale moyenne. Veuillez noter que les données sont linéaires, et que seul un autoajustement a été appliqué à la visualisation.

de l'utilisateur qu'il fournisse des données complexes.
La formule de la transmittance de Rayleigh de l'atmosphère en fonction de la longueur d'onde $lambda$ nm, de la hauteur de l'observateur H m et de la pression p hPa est la suivante :
$\tau_R(\lambda, H, p) = \left(\frac{p}{1013.25}\right)\left(0.00864 + 6.5 \times 10^{-6} \cdot H\right)\lambda^{-(3.916+0.074\lambda+\frac{0.050}{\lambda})}$.
Dans des circonstances normales, la diffusion des aérosols a une réponse à peu près plate dans le domaine visible. Cela change dans des conditions spécifiques, par exemple lorsque la concentration de particules de fumée dans l'atmosphère est élevée après des incendies de forêt ou, dans certaines parties de l'Europe, lorsque la poussière du Sahara est transportée dans l'atmosphère. Toutefois, ces effets sont très difficiles à modéliser avec précision, car ils dépendent de la concentration de sable ou de particules de fumée dans l'atmosphère à ce moment-là. Siril ne modélise donc pas cet effet.
Les principales raies d'absorption moléculaire dans le spectre visible sont les bandes d'ozone stratosphérique de Chappuis et la raie d'absorption de l'oxygène moléculaire de Fraunhofer B. Cependant, la raie Fraunhofer B est très étroite et n'a pas d'effet significatif sur l'étalonnage global. Les bandes de Chappuis sont larges mais avec un faible pic d'absorption, ce qui a un impact global beaucoup plus faible que la diffusion de Rayleigh. Les bandes d'absorption moléculaires ne sont pas modélisées dans Siril.

Lorsque vous cochez la case *Correction atmosphérique*, les options suivantes sont disponibles :

— Hauteur de l'observateur. Cette option permet de définir la hauteur de l'observateur, qui est utilisée dans le calcul de l'extinction de Rayleigh. Réglez-la à l'altitude de votre observatoire au-dessus du niveau de la mer. Certains logiciels de capture définissent le mot clé `SITEELEV` dans l'en-tête FITS : si elle est présente, la hauteur de cette carte sera utilisée, sinon la valeur est modifiable et fixée par défaut à 10 m.

— Pression atmosphérique. Cette fonction permet de définir la pression atmosphérique au moment de l'observation. Pour plus de commodité, elle peut être spécifiée en tant que pression au niveau de la mer (telle que fournie par les prévisions météorologiques) ou en tant que pression locale (telle que mesurée par un baromètre à l'observatoire). En cas de doute, la valeur par défaut est la pression atmosphérique standard au niveau de la mer (1013,25 hPa).

Théorie

Si la pression est fournie en tant que mesure de la pression au niveau de la mer, la pression locale à la hauteur de l'observateur est calculée selon la formule barométrique :
$P(h) = P_0 \left(1 - \frac{Lh}{T_0}\right)^{\frac{gM}{RL}}$,
où :
- $L = 0.0065$ K/m (Gradient thermique vertical),
- $T_0 = 288.15$ K (température standard au niveau de la mer),
- $g = 9.80665$ m/s² (Accélération de la gravité),
- $M = 0.0289644$ kg/mol (masse molaire de l'air terrestre),
- $R = 8.3144598$ J/(molK) (Constante universelle des gaz).

— Airmass. Ce paramètre n'est pas modifiable, mais il indique la masse d'air qui sera utilisée dans les calculs. Elle est obtenue, par ordre de préférence, à partir du mot clé d'en-tête FITS `AIRMASS`, par calcul en utilisant le mot clé d'en-tête FITS `CENTALT` ou, en dernier recours, en utilisant l'angle zénithal moyen de toutes les parties du mot clé d'en-tête FITS situées à plus de 30° au-dessus de l'horizon. L'infobulle indique la source sur laquelle est basée la valeur utilisée.

Théorie

Si le mot clé `AIRMASS` n'est pas disponible, le calcul utilisé pour dériver la masse d'air de l'angle zénithal est calculé conformément à [Young1994] :
$X(z) = \frac{1.002432\cos^2 z + 0.148386\cos z + 0.0096467}{\cos^3 z + 0.149864\cos^2 z + 0.0102963\cos z + 0.000303978}$.

L'interface vous permet de visualiser les détails du capteur, du filtre et de la référence de blanc sélectionnés en utilisant le bouton *Détails* situé à côté de chaque liste déroulante. Dans la boîte d'informations détaillées qui s'affiche, vous avez également la possibilité de tracer l'efficacité quantique (pour les capteurs), la transmittance (pour les filtres) ou le nombre relatif de photons (pour les références de blanc) en fonction de la longueur d'onde. Un bouton *Tout tracer* est aussi disponible dans la boîte de dialogue principale SPCC, ce qui vous permet de voir les réponses de tous vos filtres et de votre capteur, ainsi que le spectre de référence de blanc, tous tracés ensemble.

Lorsque vous êtes satisfait, cliquez sur *Appliquer* et SPCC s'exécutera. Il mettra en cache les données du catalogue, mais la première fois que vous l'appliquerez à une image, il faudra quelques secondes pour effectuer les recherches dans le catalogue en ligne et récupérer la source et les données spectrales. SPCC sera ensuite appliquée à l'image. Des tracés supplémentaires montrant l'ajustement linéaire des rapports Rouge/Vert et Bleu/Vert du catalogue aux rapports Rouge/Vert et Bleu/Vert de l'image peuvent être affichés.

Astuce

Comment traiter les images L-RGB ? Nous recommandons de ne traiter que les images RVB avec SPCC. La couche L doit être ajoutée ultérieurement, lorsque les histogrammes ont été étirés.

Astuce

Pour les images prises avec un capteur couleur, nous recommandons d'utiliser le*Bayer Drizzle* pour récupérer les couleurs de l'image. Cela permet d'obtenir des couleurs plus précises, comme le montre l'image suivante.

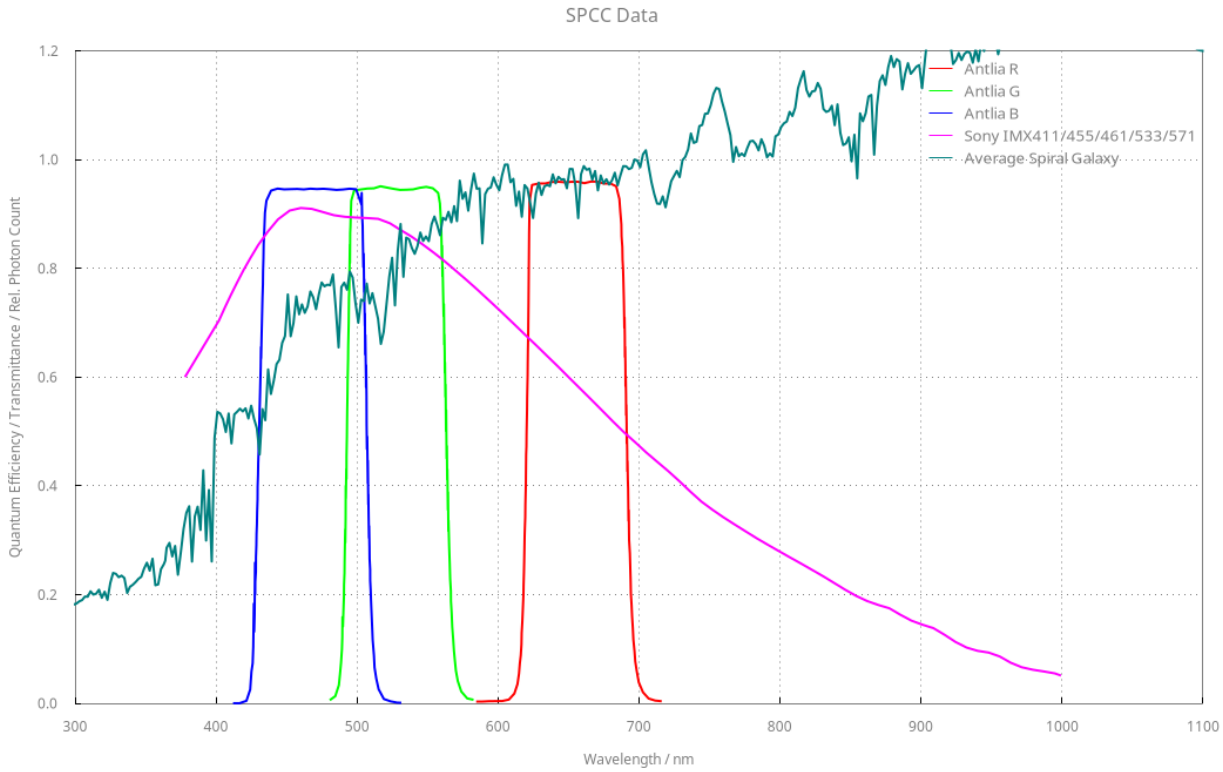


FIG. 21 – Tracé de toutes les réponses de tous vos filtres, de votre capteur et du spectre de référence de blanc

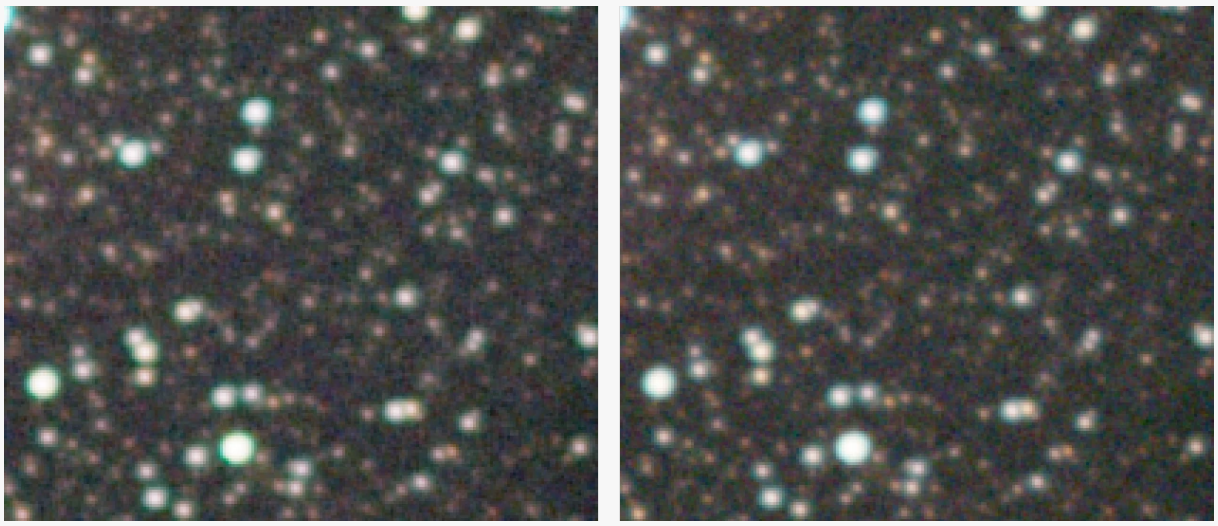


FIG. 23 – SPCC appliqué à l'identique à la même image. À gauche, le dématriçage conventionnel utilisant l'algorithme VNG ; à droite, la technique Bayer Drizzle. Une teinte verte dominante est clairement visible sur l'image dématriçée de manière conventionnelle. **Notez** que l'algorithme VNG a été choisi pour cet exemple parce que les effets expliqués ici sont plus prononcés. Cependant, dans Siril, l'algorithme de dématriçage par défaut est RCD. Cliquez sur l'image pour l'agrandir.

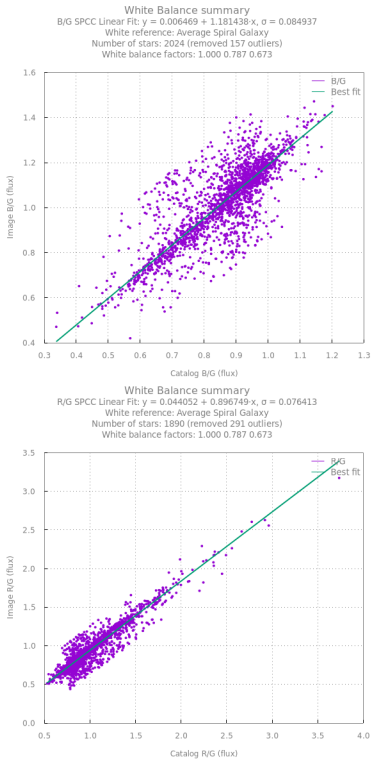


FIG. 22 – Par défaut, Siril produit des graphiques montrant les ajustements utilisés dans le processus. Dans cet exemple, la magnitude a été limitée à 17.

Base de données des filtres et capteurs SPCC

Conversion des données

Le format utilisé pour la base de données est JSON (un format d'échange de données léger dérivé de la notation d'objets JavaScript). Nous vous recommandons de commencer par un fichier existant de la base de données qui correspond à vos besoins et de l'enregistrer sous le nom de votre capteur ou de votre filtre. Il vous suffira ensuite de remplacer les valeurs du fichier par les données que vous avez obtenues.

- Dans le tableau `wavelength`, entrez vos mesures de longueur d'onde. Veillez à ce que le champ `units` soit correctement réglé sur l'une des valeurs suivantes : `angstroms`, `nm`, `micrometres` ou `m`.
- Dans le tableau `values`, entrez l'un ou l'autre :
  - Valeurs de transmittance pour les filtres
  - Valeurs d'efficacité quantique pour les capteurs

  Définissez le champ `range` en fonction de l'échelle de vos données (par exemple, `"range" : 100` si vos valeurs sont des pourcentages, `range` : 1`` si elles sont normalisées à 1).

Comment contribuer

La base de données SPCC est conçue pour stocker des fichiers JSON de capteurs et de filtres OSC/monochromes disponibles sur le marché. Son objectif principal est de rassembler des données complètes, en encourageant la collaboration au sein de la communauté.

Nous apprécions grandement les contributions de la communauté et encourageons une participation active. Nous avons besoin de données couvrant idéalement une plage de 300nm à 1100nm. Des outils logiciels peuvent être utilisés pour extraire des courbes/graphiques trouvés en ligne, et il est également possible de contacter directement les fabricants pour obtenir des données.

Read this page and help us by contributing.

Important

Nous n'incluons pas les filtres à bande étroite. Ces filtres très spécifiques sont synthétisés dans Siril, ce qui garantit leur précision. Ceci s'applique également aux filtres à double bande étroite.

Référence du format de fichier JSON

Voici le modèle des fichiers JSON utilisés dans la base de données SPCC :

```
[
  {
    "model": "sensor model / filter set",
    "name": "sensor / filter name",
    "type": "MONO_SENSOR | OSC_SENSOR | MONO_FILTER | OSC_FILTER | OSC_LPF | WB_REF",
    "dataQualityMarker": 1 - 5,
    "dataSource": "Describe where the data came from",
    "manufacturer": "Manufacturer name",
    "version": 1,
    "channel": "RED | GREEN | BLUE | LUM",
    "wavelength": [Comma separated array of wavelengths],
    "values": [Comma separated array of values]
  }
]
```

Notes Importante

— Définition du champ `dataQualityMarker` :

1. Données de provenance inconnue. **Non acceptées** pour le dépôt siril-spcc-database.
2. Données numérisées à partir de graphiques du fabricant d'origine (OEM) ou d'autres graphiques réputés au format image.
3. Données tabulées de résolution inférieure fournies par l'OEM, ou données académiques relatives à la transmittance de filtres standard idéaux (par exemple, filtres photométriques standard génériques).
4. Données tabulées à haute résolution (pas plus de 2nm d'espacement) fournies par l'OEM.
5. Données spécifiques à votre propre filtre que vous avez personnellement calibré à l'aide d'équipements appropriés. Il s'agit du marqueur de qualité le plus élevé possible et ne sera jamais attribué aux fichiers .json dans le dépôt qui ne peuvent être que génériques à un modèle d'équipement, et non spécifiques à votre élément d'équipement individuel. Notez que la qualité réelle de ces données dépend entièrement de la qualité de votre équipement de calibration - l'adage "garbage in, garbage out" (déchets en entrée, déchets en sortie) s'applique.

— Exigences pour le nom du `model` :
  — Doit être identique pour tous les objets JSON liés dans un ensemble
  — Exemples :
    — Ensemble de filtres RGB : `"model": "Chroma RGB"`
    — Capteur OSC : `"model": "ZWO ASI2600MM"`
— Le champ `channel` :
  — Requis uniquement pour `"type": "OSC_SENSOR"` ou `"type": "MONO_FILTER"`
  — Pour les capteurs OSC, incluez un objet JSON par canal (`RED`, `GREEN`, `BLUE`)
  — Ordre préféré des canaux : `RED`, `GREEN`, `BLUE`
— Exigences pour le tableau `wavelength` :

— Couverture minimale : 380nm à 700nm
— Plage utile maximale : 336nm à 1020nm (limites spectrales Gaia DR3)
— Les valeurs doivent être strictement croissantes
— Aucune valeur en double n'est autorisée
— Doit utiliser les unités spécifiées (`angstroms`, `nm`, `micrometres`, `m`)

> **Note**
>
> Si vos données de capteur ne s'étendent que jusqu'à 400nm (ce qui est courant avec certains fabricants), il est acceptable d'extrapoler un point unique à 380nm. La réponse du capteur en dessous de 400nm suit généralement un modèle prévisible sur différents capteurs. Ajouter ce point extrapolé à 380nm est préférable à laisser la courbe se terminer à 400nm, ce qui traiterait effectivement toute réponse en dessous de 400nm comme nulle. L'impact de cette extrapolation est minime car la réponse CIE 1931 est très faible dans cette plage de longueurs d'onde.

— Exigences pour le tableau `values` :
  — Pour les filtres : contient des valeurs de transmittance
  — Pour les capteurs : contient des valeurs d'efficacité quantique
  — Définir une valeur `range` appropriée (par ex., 100 pour les pourcentages)
  — Siril met à l'échelle toutes les valeurs dans une plage de 0.0-1.0 en interne

Préférences enregistrées

Comme la plupart des utilisateurs sont susceptibles d'effectuer la majeure partie de leur travail d'imagerie avec une seule configuration, voire deux, il serait fastidieux de sélectionner à nouveau le capteur et les filtres à chaque fois. Les choix de l'utilisateur sont donc automatiquement mémorisés lorsqu'ils sont définis et restaurés lors de la prochaine utilisation de l'outil, même si Siril est fermé et redémarré entre-temps. Cela fonctionne avec le système de préférences, mais il n'est pas nécessaire d'utiliser la boîte de dialogue des préférences pour se souvenir du capteur et des filtres choisis, cela se fait automatiquement.

La référence de blanc choisie n'est pas mémorisée : la référence par défaut, Moyenne des Galaxies spirales, est un choix approprié pour la plupart des scènes astronomiques, et d'autres références de blanc seraient normalement définies pour une image spécifique afin de faire ressortir un aspect particulier de la couleur.

> **Ligne de commande Siril**

```
spcc [-limitmag=[+-]] [ { -monosensor= [ -rfilter= ] [-gfilter=] [-bfilter=] | -
↪oscsensor= [-oscfilter=] [-osclpf=] } ] [-whiteref=] [ -narrowband [-rwl=] [-gwl=]␣
↪[-bwl=] [-rbw=] [-gbw=] [-bbw=] ] [-bgtol=lower,upper] [ -atmos [-obsheight=] { [-
↪pressure=] | [-slp=] } ]
```

Exécute la Correction de Couleur par Spectrophotométrie sur l'image chargée et résolue astrométriquement.

La magnitude limite des étoiles est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement une valeur positive absolue pour la magnitude limite.
Le catalogue d'étoile utilisé pour SPCC est toujours Gaia DR3 : par défaut le catalogue local Gaia DR3 xp_sampled est utilisé si disponible mais cela peut être remplacé avec **-catalog={gaia | localgaia}**.

Les noms des capteurs et des filtres peuvent être spécifiés à l'aide des options suivantes : **-monosensor=**, **-rfilter=**, **-gfilter=**, **-bfilter=** ou **-oscsensor=**, **-oscfilter=**, **-osclpf=** ; le nom de la référence des blancs peut être spécifié à l'aide de l'option **-whiteref=**. Dans tous les cas, le nom doit être fourni exactement tel qu'il apparaît dans les listes

déroulantes de l'outil SPCC. Notez que les noms des capteurs, des filtres et des références blanches peuvent contenir des espaces : dans ce cas, lorsqu'ils sont utilisés comme arguments de la commande **spcc**, l'argument entier doit être placé entre guillemets, par exemple "-whiteref=Average Spiral Galaxy".

Le mode bande étroite peut être sélectionné à l'aide de l'argument **-narrowband**, auquel cas les arguments de filtre précédents sont ignorés et les longueurs d'onde et largeurs de bande du filtre peuvent être fournies à l'aide de **-rwl=**, **-rbw=**, **-gwl=**, **-gbw=**, **-bwl=** et **-bbw=**.

Si l'un des arguments des données spectrales est omis, la valeur utilisée précédemment sera utilisée.

Background reference outlier tolerance can be specified in sigma units using **-bgtol=lower,upper** : these default to -2.8 and +2.0.

La correction atmosphérique peut être appliquée en passant **-atmos**. Dans ce cas, les arguments facultatifs suivants s'appliquent : **-obsheight=** spécifie la hauteur de l'observateur au-dessus du niveau de la mer en mètres (10 par défaut), **-pressure=** spécifie la pression atmosphérique locale sur le site d'observation en hPa, ou **-slp=** spécifie la pression atmosphérique au niveau de la mer en hPa (la pression par défaut est de 1013,25 hPa au niveau de la mer)

```
spcc_list { oscsensor | monosensor | redfilter | greenfilter | bluefilter | oscfilter␣
↪| osclpf | whiteref }
```

Imprime une liste de noms SPCC pouvant être utilisés pour définir des capteurs, des filtres ou des références de blanc à l'aide de la commande **spcc**. Cette commande nécessite un argument pour définir la liste à imprimer : les options sont **oscsensor**, **monosensor**, **redfilter**, **greenfilter**, **bluefilter**, **oscfilter**, **osclpf** ou **whiteref**.
Notez que les noms des capteurs, des filtres et des références de blanc peuvent contenir des espaces : dans ce cas, lorsqu'ils sont utilisés comme arguments de la commande **spcc**, l'argument entier doit être placé entre guillemets, par exemple "-whiteref=Average Spiral Galaxy"

Liens : *spcc*

Références

10.3 Autres outils de couleur

10.3.1 Saturation des couleurs

Cet outil est utilisé pour augmenter la saturation des couleurs de l'image. Il est possible de choisir entre une teinte spécifique ou la teinte globale à rehausser. L'intensité de la saturation est ajustée à l'aide du curseur *Quantité*.

Le curseur *Facteur de fond de ciel* définit le facteur multiplicatif pour la valeur du fond du ciel. Plus la valeur est faible, plus l'effet de saturation est important. Une valeur élevée préservera le fond du ciel.

Ligne de commande Siril

```
satu amount [background_factor [hue_range_index]]
```

Renforce la saturation des couleurs de l'image chargée. Essayez de manière itérative pour obtenir les meilleurs

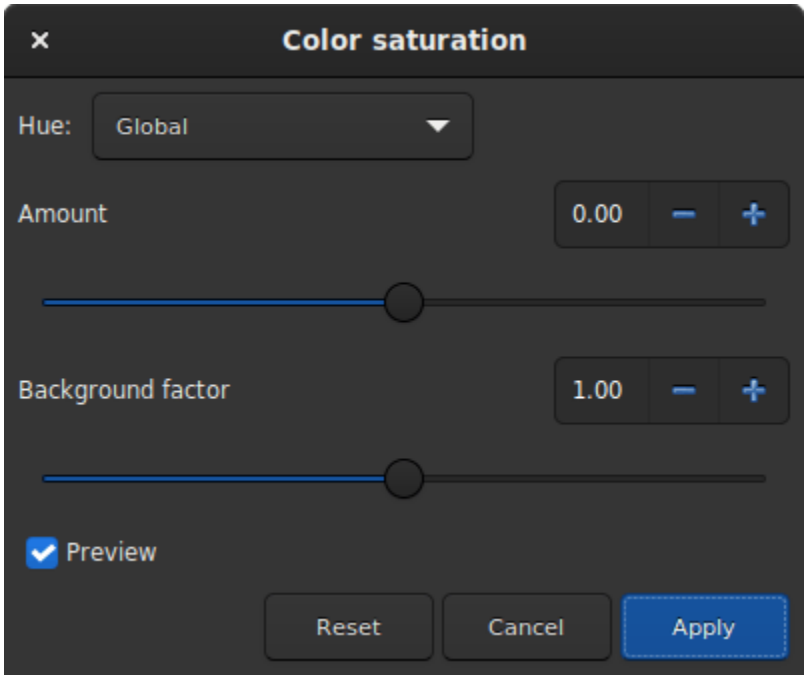


FIG. 24 – Fenêtre de dialogue Saturation des couleurs.

résultats.

amount peut être un nombre positif pour augmenter la saturation des couleurs, négatif pour la réduire, 0 ne fait rien, 1 l'augmente de 100%

background_factor est un facteur de (médiane + écart-type) utilisé pour définir un seuil, de sorte que seuls les pixels au-dessus de ce seuil seraient modifiés. Cela permet d'avoir un bruit de fond non saturé en couleur, si le choix est fait avec soin. La valeur par défaut est de 1. Une valeur de 0 désactive le seuil.

hue_range_index peut être entre [0, 6], ce qui signifie : 0 pour rose à orange, 1 pour orange à jaune, 2 pour jaune à cyan, 3 pour cyan, 4 pour cyan à magenta, 5 pour magenta à rose, 6 pour tout (par défaut)

10.3.2 Suppression du bruit vert

Le vert n'étant pas naturellement présent dans les images du ciel profond (à l'exception des comètes et de certaines nébuleuses planétaires), si l'image a déjà été calibrée, que ses couleurs sont bien équilibrées et qu'elle ne présente aucun gradient, on peut supposer que si l'image contient du vert, c'est que cela fait partie du bruit. Il est alors intéressant de trouver une méthode pour supprimer ce vert dominant. C'est exactement ce que propose l'outil Suppression du bruit vert, qui est dérivé de l'outil Subtractive Color Noise Reduction de Pixinsight, mais pour le vert uniquement.

Avertissement

Cet outil n'est pas destiné à être utilisé directement sur l'image issue de l'empilement, avec une teinte verte typique, auquel le niveau de l'arrière-plan n'a pas été égalisé. Son utilisation dans de telles conditions détruirait la chrominance de l'image.

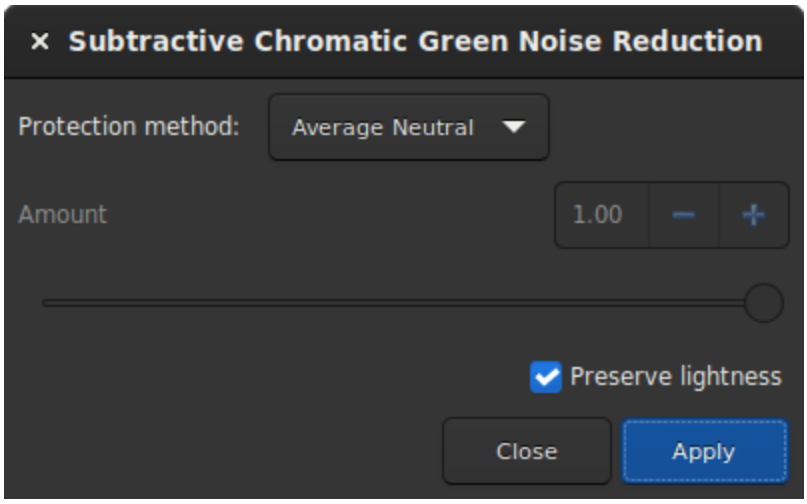


FIG. 25 – Fenêtre de dialogue Suppression du bruit vert.

Avertissement

Cet outil est conçu pour être utilisé sur des images non linéaires, car il fonctionne dans un espace colorimétrique non linéaire. Assurez-vous que l'histogramme a été étiré avant d'utiliser cet outil afin d'éviter des résultats de traitement incorrects.

Cet outil a 3 paramètres. La méthode de protection, le montant (appelé a dans la section suivante), et un bouton *Préserve la luminosité*. Les méthodes suivantes présentent les différentes façons existantes de supprimer les pixels verts en les remplaçant par un mélange de rouge et de bleu. La quantité n'est disponible que pour les méthodes avec protection par masque. Le choix de sa valeur doit être fait avec prudence afin de minimiser la montée de la dominante magenta dans le fond du ciel. N'hésitez pas à utiliser les boutons *Annuler* et *Rétablir* pour affiner la valeur.

Méthode de protection

Protection Masque maximum

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

Protection Masque additif

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

Protection Neutre Moyen (méthode par défaut)

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

Protection Neutre Maximum

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

Finalement le bouton *Préserver la luminosité* préserve la composante CIE L* originale dans l'image traitée, afin de ne traiter que la composante chromatique, il est fortement recommandé de laisser cette option cochée.

Ligne de commande Siril

```
rmgreen [-nopreserve] [type] [amount]
```

Applique un filtre de réduction de bruit chromatique. Il supprime la teinte verte de l'image actuelle. Ce filtre est basé sur SCNR de PixInsight et c'est aussi le même filtre utilisé par le plugin HLVG de Photoshop.
La luminosité est préservée par défaut mais peut être désactivé avec le paramètre **-nopreserve**.

Type peut prendre les valeurs 0 pour neutre moyen, 1 pour neutre maximum, 2 pour masque maximum, 3 pour masque additif, par défaut à 0. Les deux derniers peuvent prendre un argument **amount**, une valeur entre 0 et 1, la valeur par défaut étant 1

10.3.3 Transformation négative

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.

L'outil *Transformation négative* est différent de la vue négative dans la barre d'outil. En effet, la transformation n'est pas seulement visuelle, elle s'applique également aux valeurs des pixels. Si vous enregistrez l'image, elle sera enregistrée en négatif.

Astuce

Une utilisation courante de l'outil de transformation négative est la suppression de la dominante magenta des images SHO. Dans ce cas, il faut appliquer *Transformation négative*, puis *Suppression du bruit vert*, puis *Transformation négative* à nouveau.

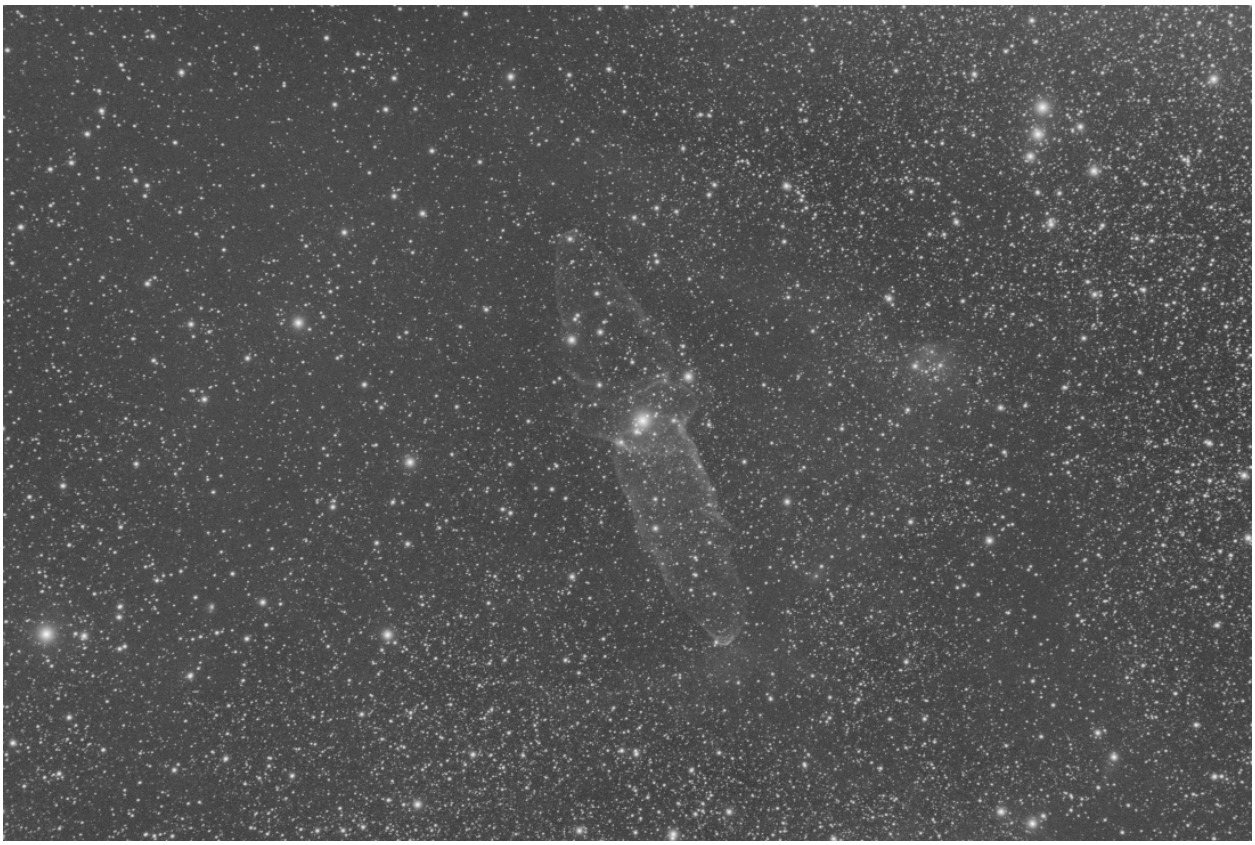


FIG. 26 – Image originale avec signal faible (Image Cyril Richard).

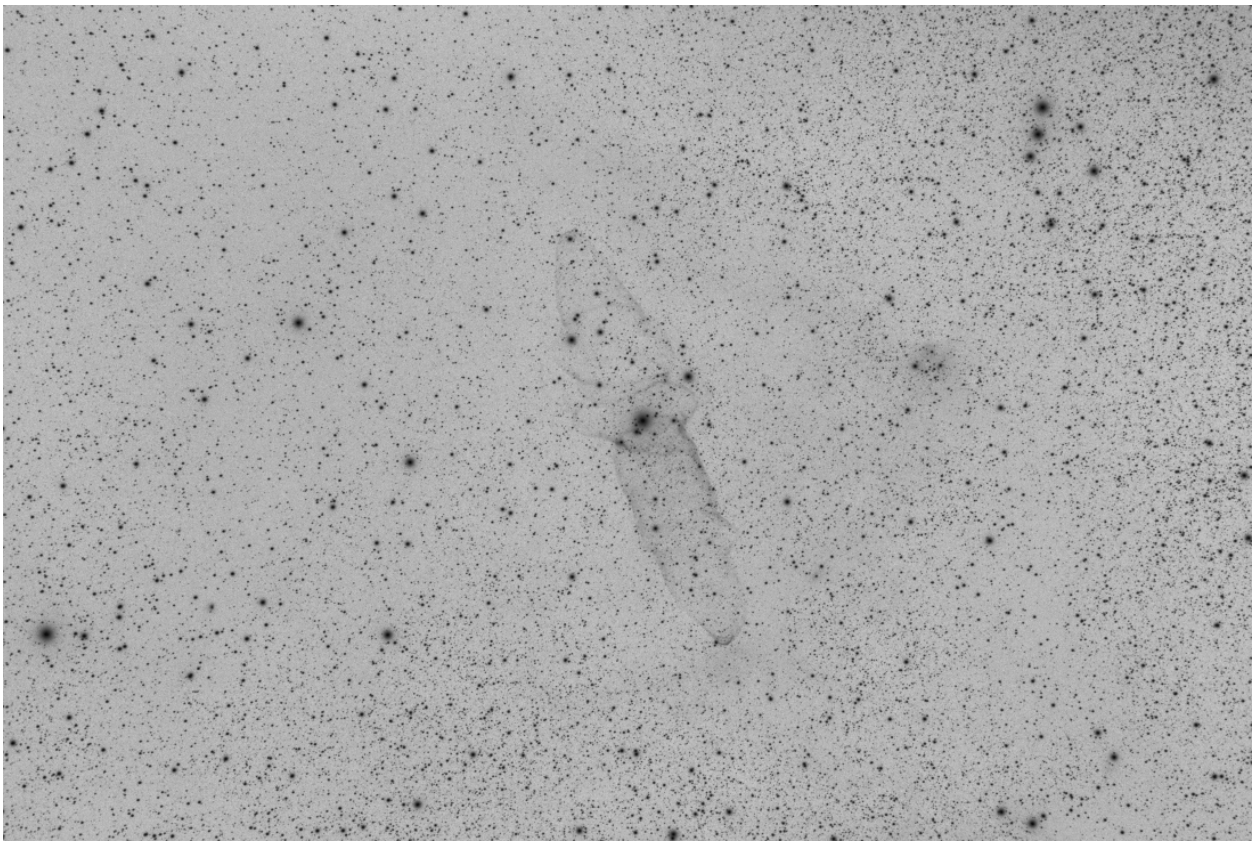


FIG. 27 – Image négative où le signal est plus visible (Image Cyril Richard).

Ligne de commande Siril

```
neg
```

Modifie les valeurs de pixels de l'image actuellement chargée en une vue négative, comme 1-valeur pour 32 bits, ou 65535-valeur pour 16 bits. Cela ne modifie pas le mode d'affichage

10.4 Filtres

Cette section présente tous les filtres présents dans Siril. Les filtres sont des outils qui vont modifier les pixels de l'image en fonction des besoins.

10.4.1 Transformation d'ondelettes à trous

Une ondelette est une fonction à la base de la décomposition en ondelettes, une décomposition similaire à la transformée de Fourier à court terme, utilisée dans le traitement des signaux. Elle correspond à l'idée intuitive qu'une fonction correspondant à de petites oscillations, d'où son nom.

Il existe de nombreux types de fonctions d'ondelettes qui ont leurs propres noms, comme le montre la figure ci-dessous.

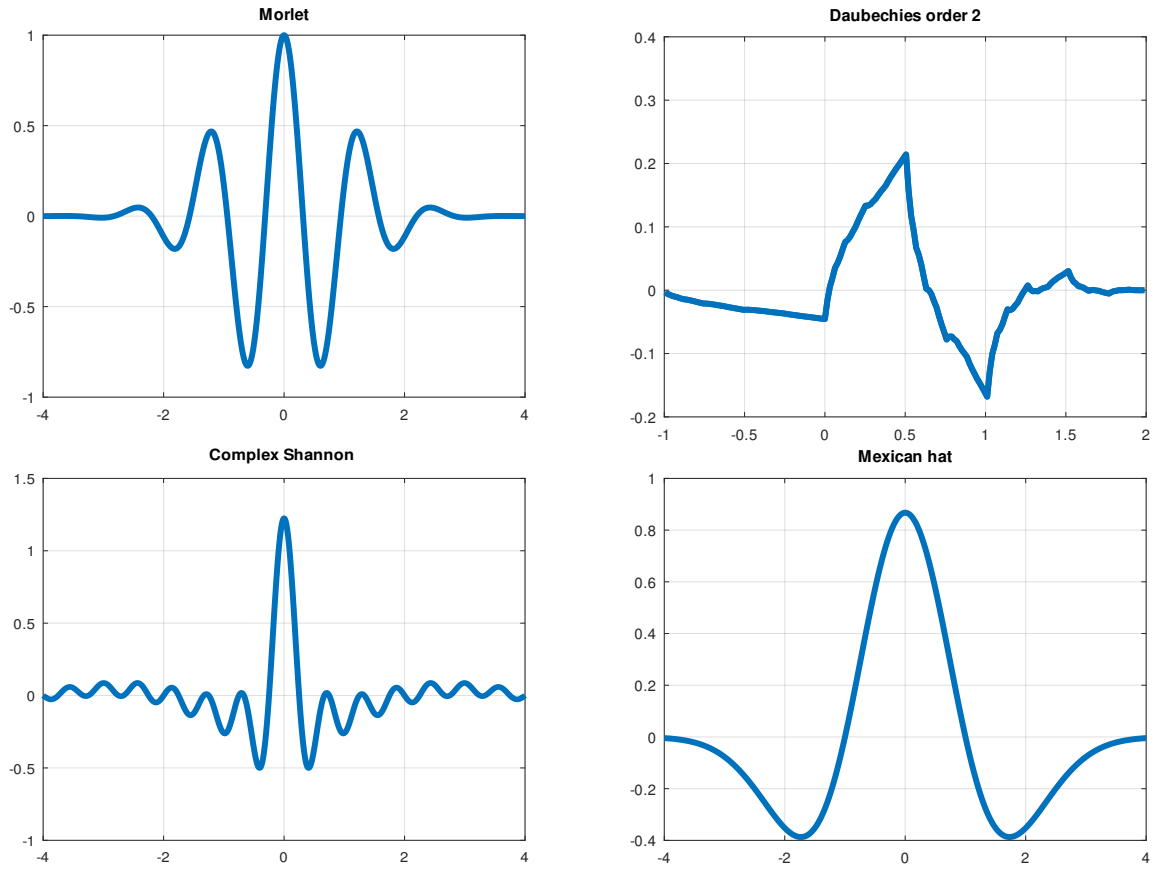


Fig. 28 – Un exemples des quatre différents types d'ondelettes.

La transformée en ondelettes À Trou utilisée dans Siril décompose une image en une série de couches d'échelle, éga-

lement appelées couches d'ondelettes.

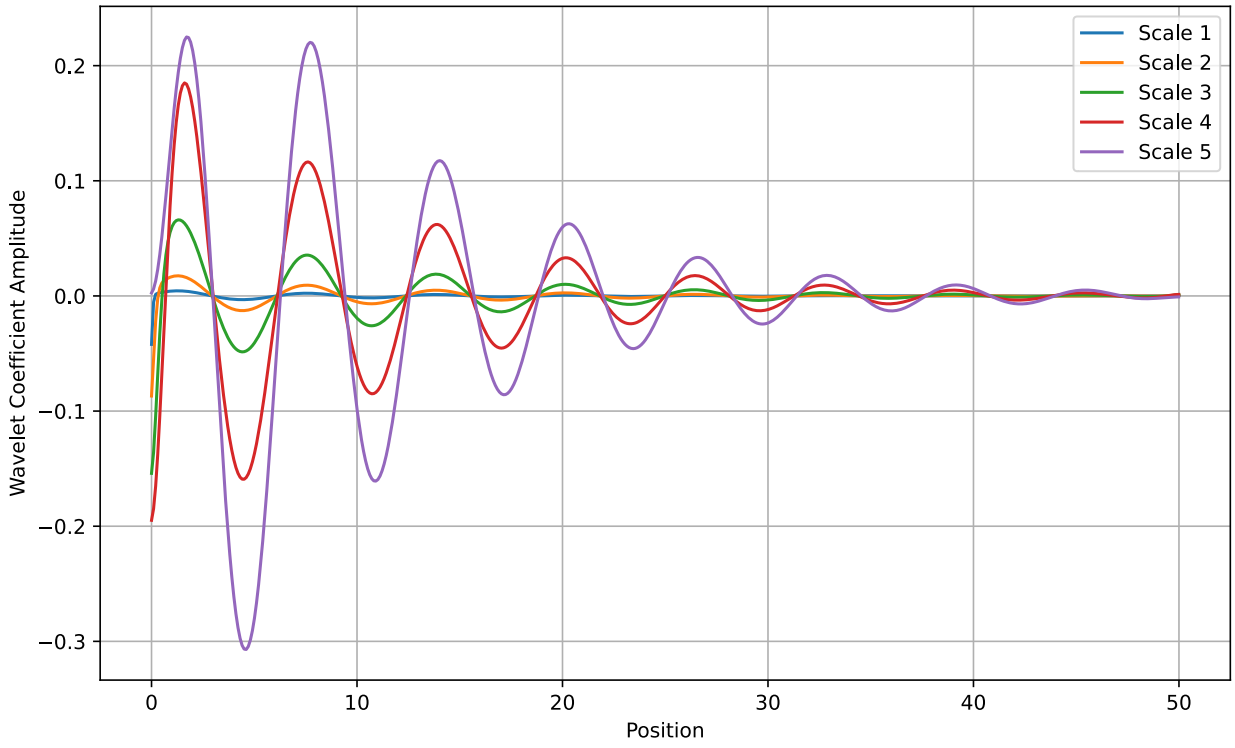


FIG. 29 – Représentation par transformée d'ondelettes à Trous avec interpolation bspline.

Ces couches peuvent être extraites avec l'outil d'extraction *Couches des ondelettes*, cependant ici, elles sont utilisées sans être visuellement accessibles. En général, cet algorithme est largement utilisé à la fin d'un empilement d'images planétaires. Le bruit étant exclusivement contenu dans l'une des couches d'ondelettes, il est possible de faire ressortir les détails de l'image en limitant la montée de bruit.

La première chose à faire est de cliquer sur le bouton*Exécuter* afin de calculer les couches d'ondelettes en utilisant les paramètres définis ci-dessus, tels que :

— **Type** : Il y a deux type d'algorithmes possible : Linear et BSpline. Ce dernier sera le plus souvent choisi, même si il est plus lent.

— **Nb de couches** : Nombre de couches d'ondelettes qui seront utilisées. 6 est le nombre maximal de couches pouvant être défini. Pour travailler sur un plus grand nombre de couches il est possible d'utiliser la ligne de commande expliquée ci-dessous.

Ensuite, chaque couche a un curseur qui permet de modifier le contraste de cette couche. Si moins de 6 couches ont été créées, alors seuls les curseurs correspondants seront actifs. Une valeur supérieure à 1 améliore les détails tandis qu'une valeur plus petite a tendance à les réduire.

Ceci est un outil de visualisation en temps réel. Les changements sont affichés en temps réel et vous devez cliquer sur *Appliquer* pour les valider. En cliquant sur *Réinitialiser*, tous les curseurs sont remis à 1, annulant ainsi toute transformation en cours.

Ligne de commande Siril

```
wavelet nbr_layers type
```

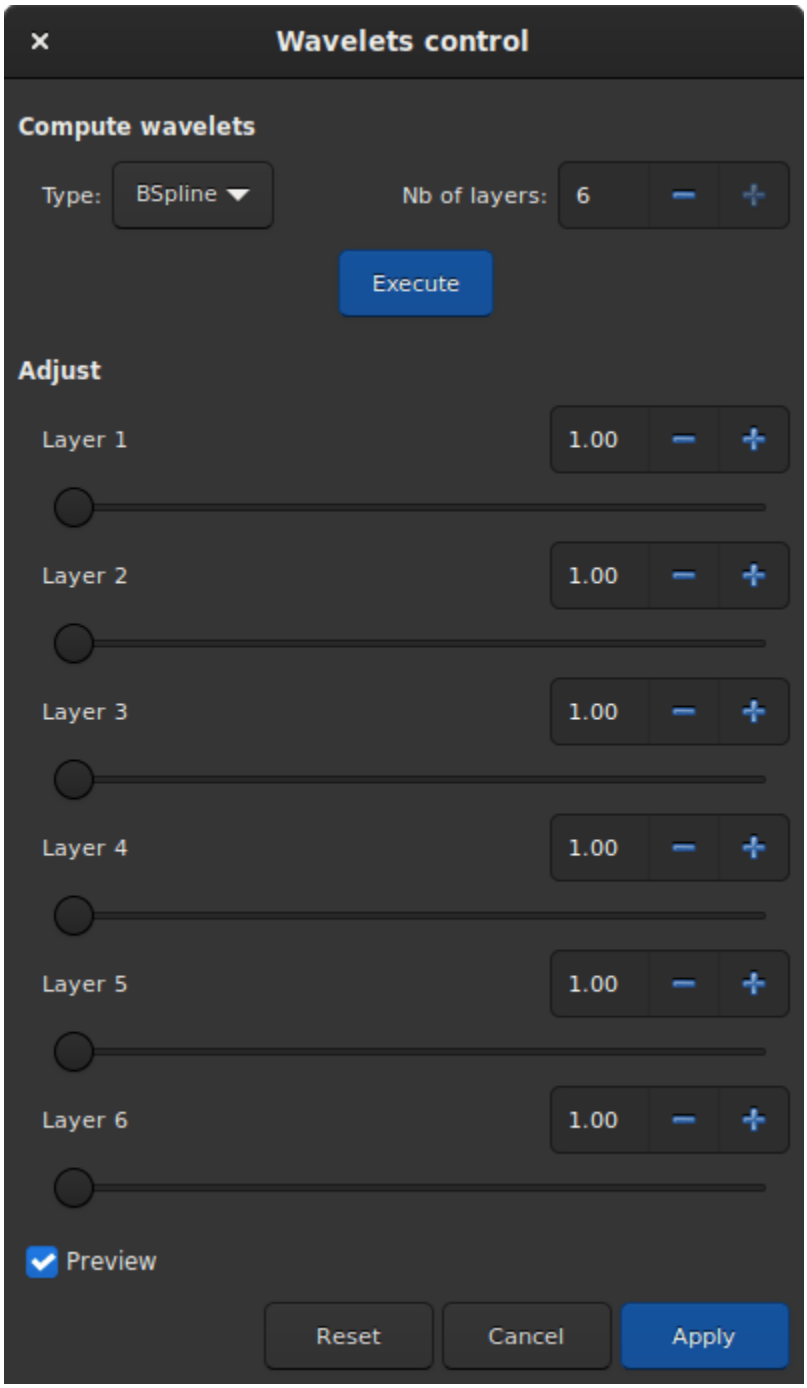


FIG. 30 – Boite de dialogue de l'outil ondelette.

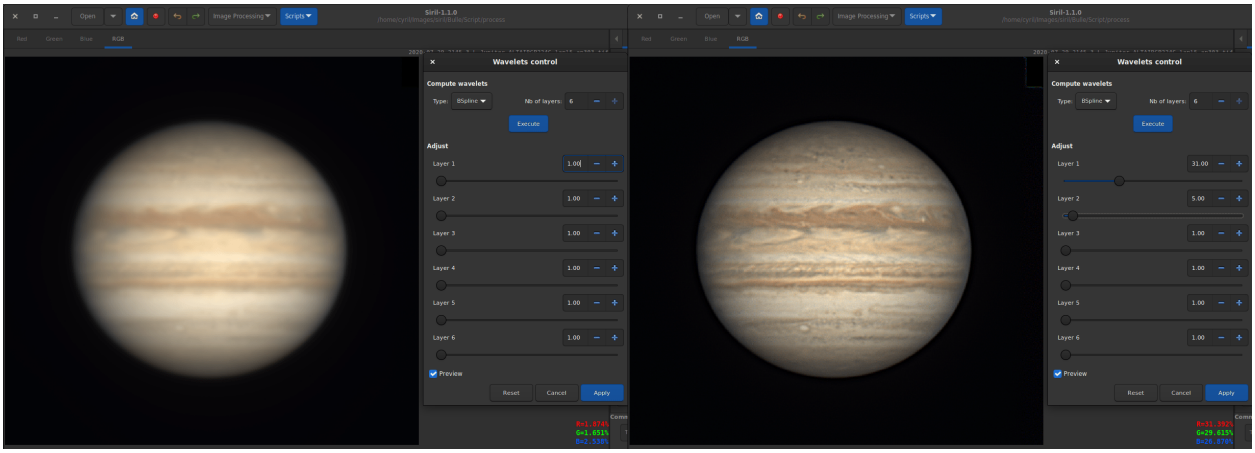


FIG. 31 – Ondelettes appliquées à une image de Jupiter (avec l'aimable autorisation de J.-L. Dauvergne). L'image de gauche est l'image brute à la sortie de l'empilement, tandis que l'image de droite est la même image sur laquelle les ondelettes sont appliquées.

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

Voir aussi EXTRACT

Liens : *wrecons*, *extract*

Ligne de commande Siril

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

Reconstruit l'image actuelle à partir des couches précédemment calculées avec les ondelettes et pondérées avec les coefficients **c1**, **c2**, ..., **cn** en fonction du nombre de couches utilisées pour la transformation en ondelettes, après l'utilisation de WAVELET

Liens : *wavelet*

L'exemple donné dans l'image ci-dessus peut être écrit dans la ligne de commande comme suit :

```
wavelet 6 2
wrecons 31 5 1 1 1 1
```

10.4.2 Réduction de la trame

Dans certains cas, les images peuvent souffrir d'un défaut de banding. Cela est généralement causé par le capteur et la calibration avec des darks, des offsets et des flats ne permet pas d'améliorer les images.

La fenêtre de dialogue de réduction du banding comporte plusieurs paramètres permettant d'optimiser le traitement :

— **Quantité** définit l'intensité de la correction. Plus la valeur est élevée, plus la correction est forte.

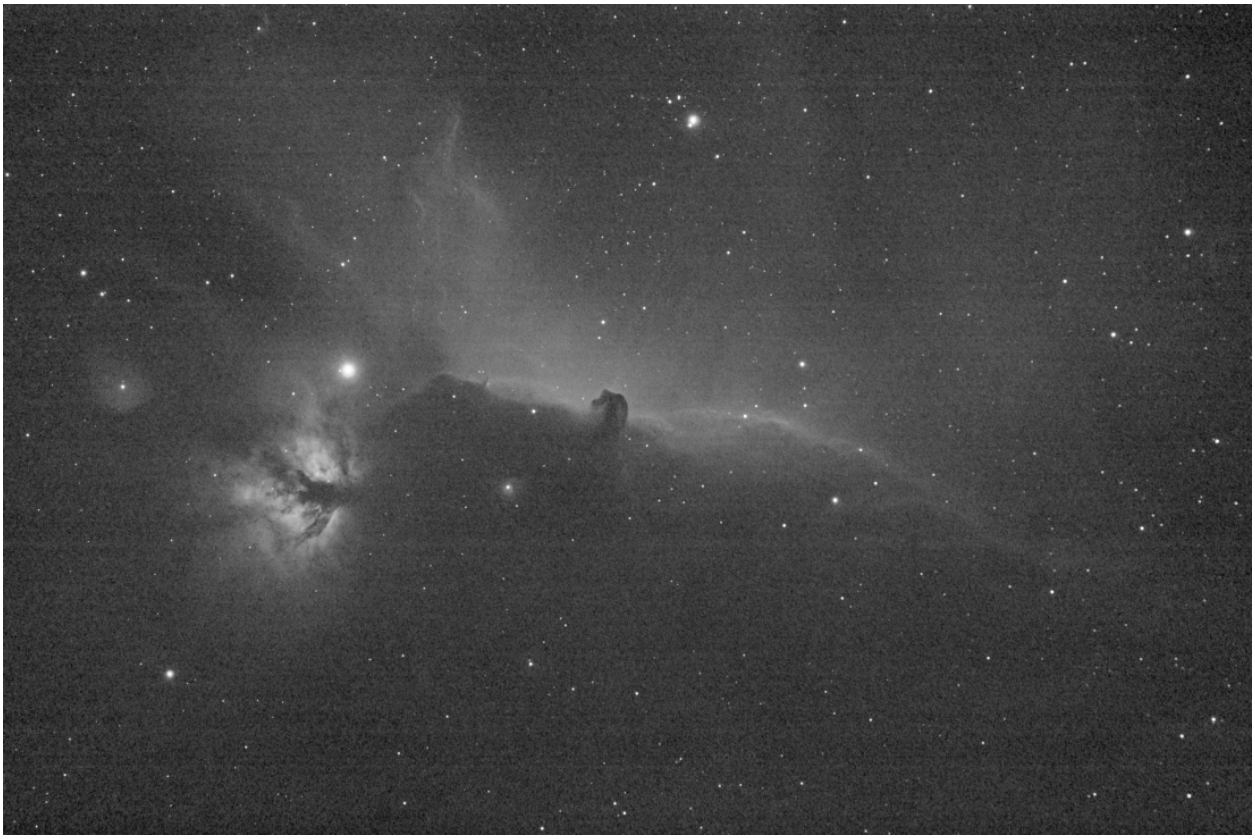


FIG. 32 – Image d'origine avec la trame visible.

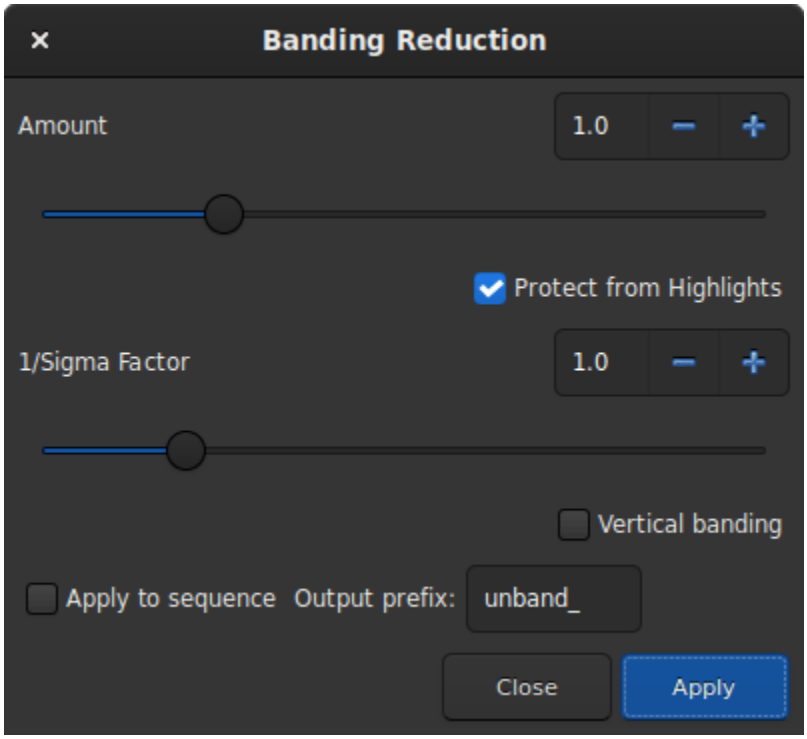


FIG. 33 – Boîte de dialogue Réduction du banding..

— **Protéger des hautes lumières** ignore les pixels lumineux lorsque l'option est cochée.
— **Coeff. 1/Sigma** permet d'ajuster la protection des hautes lumières. Une valeur plus élevée assure une meilleure protection.
— **Trame verticale** permet à l'utilisateur de corriger la trame si les bandes sont verticales.

L'application du filtre suivant à l'image originale, avec les valeurs de paramètres indiquées dans l'illustration, donne un beau résultat sans trames.

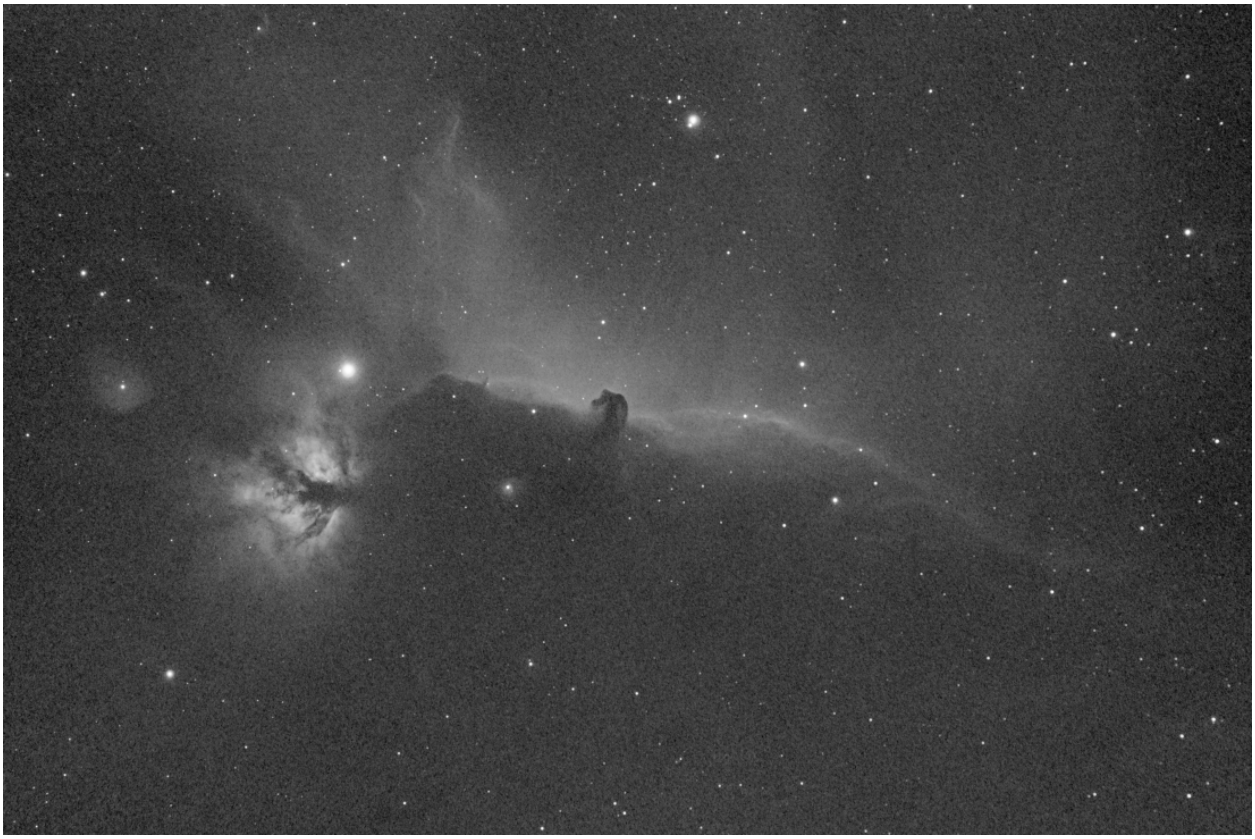


FIG. 34 – Résultat après le passage du filtre. Plus aucune trame n'est visible.

Cette transformation peut facilement être appliquée à une séquence. Il suffit de définir la transformation sur l'image chargée (avec une séquence déjà chargée), puis de cocher le bouton *Appliquer à la séquence* et de définir le préfixe de sortie de la nouvelle séquence (`unband_` par défaut).

Ligne de commande Siril

```
fixbanding amount sigma [-vertical]
```

Tente de supprimer les trames horizontales ou verticales de l'image chargée.

amount définit le montant de la correction, entre 0 et 4.

sigma définit le niveau de protection de l'algorithme, plus sigma est élevé, plus la protection est importante, entre 0 et 5. Des valeurs de 1 et 1 sont souvent suffisantes.

L'option **-vertical** permet de supprimer les trames verticales, l'option horizontale étant la valeur par défaut

Ligne de commande Siril

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

Même commande que FIXBANDING mais pour la séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "unband_", sauf indication contraire avec l'option **-prefix=**

Liens : *fixbanding*

10.4.3 Égalisation d'histogramme adaptatif limité par contraste (CLAHE)

La méthode CLAHE est utilisée pour améliorer le contraste des images. Elle diffère de l'égalisation d'histogramme ordinaire en ce que la méthode adaptative calcule plusieurs histogrammes, chacun correspondant à une section distincte de l'image, et les utilise pour redistribuer les valeurs de luminosité de l'image. Elle peut ainsi améliorer le contraste local et améliorer la définition des bords dans chaque région d'une image.

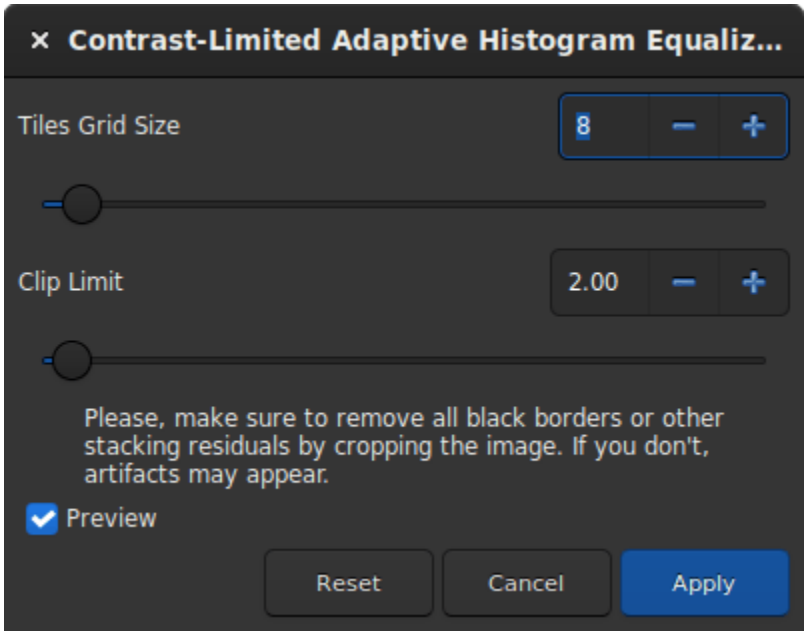


FIG. 35 – Boite de dialogue du filtre Égalisation d'histogramme adaptatif limité par contraste.

Astuce

Ce filtre est un filtre de visualisation en direct. En d'autres termes, chaque modification des paramètres est automatiquement visible à l'écran, mais cela peut être désactivé en décochant le bouton *Aperçu*.

— La taille des tuiles dans lesquelles les histogrammes sont calculés peut être définie à l'aide d'un curseur. Par défaut, elle est réglée sur 8.
— La limite de Clip est l'option qui empêche la sur-amplification du bruit dans les régions relativement homogènes d'une image. Ensuite, la partie clippée de l'histogramme qui dépasse la limite de clip est redistribuée également dans tous les compartiments de l'histogramme.

Astuce

Ce filtre fonctionne mieux sur des données non linéaires. Il est recommandé d'étirer l'image avant.

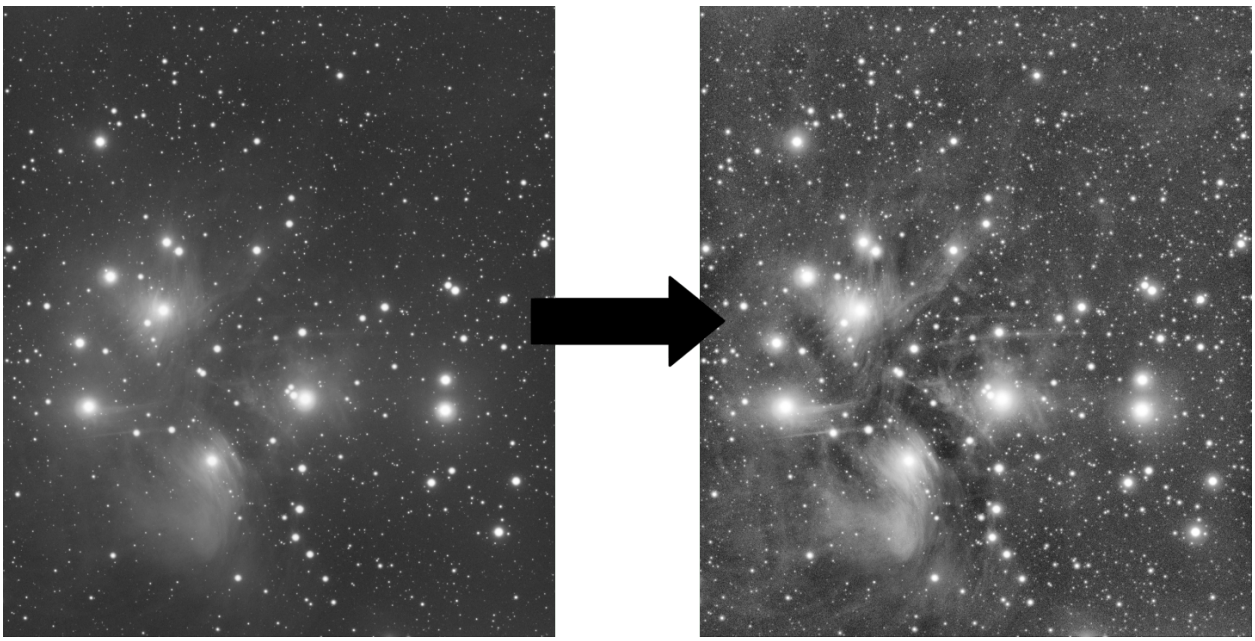


Fig. 36 – Voici un exemple de filtre CLAHE appliqué à des données non linéaires avec `Tiles Grid Size=21` et `Clip Limit=4.20`.

Ligne de commande Siril

```
clahe cliplimit tileSize
```

Égalise l'histogramme d'une image en utilisant l'égalisation d'histogramme par contraste adaptatif limité.

cliplimit définit le seuil pour la limitation du contraste.
tilesize définit la taille de la grille pour l'égalisation d'histogramme. L'image d'entrée sera divisée en tuiles rectangulaires de taille égale

10.4.4 Correction cosmétique

Dans Siril, la correction cosmétique est l'étape qui permet de se débarrasser des pixels chauds et froids de l'image. Elle est généralement effectuée lors du prétraitement à l'aide du master-dark. En effet, ce dernier contient généralement une bonne carte des pixels défectueux et il est plus facile de les trouver sur lui. Cependant, lorsque vous ne disposez pas d'un master-dark, Siril offre une alternative avec un algorithme de détection automatique de ces pixels dans une image brute.

La fenêtre de dialogue contient plusieurs paramètres nécessaires au bon fonctionnement de l'outil. Cependant, l'utilisation des paramètres par défaut donne généralement de bons résultats.

— **Sigma froid** : Combien de fois (en unités d'écart moyen) la valeur d'un pixel doit-elle différer de celle de ses voisins pour être considérée comme un pixel froid.

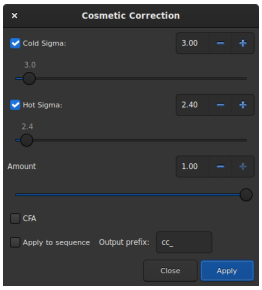


FIG. 37 – Fenêtre de dialogue de l'outil Correction cosmétique.

— **Sigma froid** : Combien de fois (en unités d'écart moyen) la valeur d'un pixel doit-elle différer de celle de ses voisins pour être considérée comme un pixel chaud.
— **Quantité** : Il s'agit d'un paramètre de modulation où 0 signifie aucune correction et 1, 100% corrigé.
— **CFA** : Cette option doit être coché pour des images CFA avec une matrice de Bayer. Ceci ne fonctionne pas avec les capteurs X-Trans.

Cette opération peut être appliquée à des séquences. Ouvrez une séquence et préparez les paramètres que vous voulez utiliser, puis cochez le bouton *Appliquer à la séquence* et définissez le préfixe de sortie de la nouvelle séquence (`cc_` par défaut).

Théorie

Détection des pixels chauds

Appelons $m_{5\times5}$ la médiane des 5 plus proches voisins Si la valeurs d'un pixel est plus grande que

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

avec avgDev, la *Déviation moyenne* de l'image entière.

Alors le pixel est remplacé par la moyenne des 3×3 pixels : $a_{3\times3}$, mais seulement si

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

Détection des pixels froids

Si la valeur du pixel est inférieure à

$$m_{5\times5} - (\sigma_{\text{low}} \times \text{avgDev}),$$

alors le pixel est remplacé par $m_{5\times5}$.

FIG. 38 – Animation montrant la correction cosmétique.

Ligne de commande Siril

```
find_cosme cold_sigma hot_sigma
```

Détection et remplacement automatique des pixels froids et chauds dans l'image chargée, avec les seuils transmis en arguments en unités sigma

Ligne de commande Siril

```
find_cosme_cfa cold_sigma hot_sigma
```

Même commande que FIND_COSME mais pour les images CFA

Liens : *find_cosme*

10.4.5 Déconvolution

La déconvolution est un outil mathématique permettant de compenser les effets de flou ou de distorsion dans une image. La véritable scène n'est pas ce qui est enregistré par votre capteur - vous enregistrez une estimation de la véritable scène convoluée par une PSF (en termes mathématiques, le "flou PSF" qui représente la distorsion atmosphérique, les propriétés physiques de votre télescope, le flou de mouvement, etc., dégradant votre estimation). La déconvolution peut, dans une certaine mesure, inverser cette dégradation d'image. Cependant, il est important de dire dès le départ que la déconvolution est ce que les mathématiciens appellent un problème mal posé (comme la plupart des problèmes inverses). Mal posé signifie qu'une solution peut ne pas exister, si une solution existe, elle peut ne pas être unique et elle peut ne pas avoir de dépendance continue par rapport aux données. Essentiellement, cela signifie que la déconvolution est, même théoriquement, très difficile et qu'il n'y a aucune garantie que cela fonctionnera.

Tout cela est encore plus difficile lorsque nous ne connaissons pas exactement la PSF que nous essayons de supprimer. En astronomie, nous pouvons théoriquement avoir une idée de la PSF grâce à l'effet de flou sur les sources ponctuelles (étoiles) que nous imagons. Cependant, parfois la véritable PSF n'est pas constante à travers l'image, parfois d'autres facteurs tels que la saturation des étoiles empêchent la PSF de l'étoile d'être une estimation entièrement précise de la PSF, et parfois (par exemple pour l'imagerie lunaire) il n'y a pas d'étoiles.

Siril vise à fournir une approche flexible de la déconvolution. Il existe plusieurs options pour définir ou estimer la PSF, et plusieurs algorithmes de déconvolution à choisir pour l'étape finale de la déconvolution une fois que la PSF est définie.

Fig. 39 – Exemple de déconvolution sur un champ d'étoiles.

La déconvolution est accessible par le menu *Traitement de l'image* ou par les commandes Siril.

Vue d'ensemble de l'utilisation

— Pour générer une PSF de déconvolution, sélectionnez la méthode de génération de PSF requise et appuyez sur *Générer PSF*. Cette opération peut être effectuée séparément de la déconvolution proprement dite afin que l'utilisateur puisse voir l'effet de la modification des paramètres de la PSF.
— Siril ne génère que des PSF monochromes car il s'agit du cas d'utilisation le plus courant et cela simplifie l'interface utilisateur. Cependant, trois PSF monochromes peuvent être sauvegardées et composées pour produire une PSF à 3 canaux qui peut être chargée et utilisée pour déconvoluer des images à 3 canaux.
— Pour appliquer la déconvolution à une seule image, sélectionnez la méthode de génération de PSF requise et appuyez sur *Appliquer*. Si une méthode d'estimation de PSF aveugle a été exécutée précédemment, la méthode sera automatiquement définie sur *PSF précédente*, afin d'éviter de recalculer inutilement la PSF.
— Pour appliquer la déconvolution à une séquence, procédez comme ci-dessus mais assurez-vous de cocher l'option *Appliquer à la séquence*. Vous pouvez également spécifier un préfixe personnalisé à donner à la séquence de sortie : si aucun autre préfixe n'est fourni, le préfixe par défaut (`dec_`) sera utilisé.

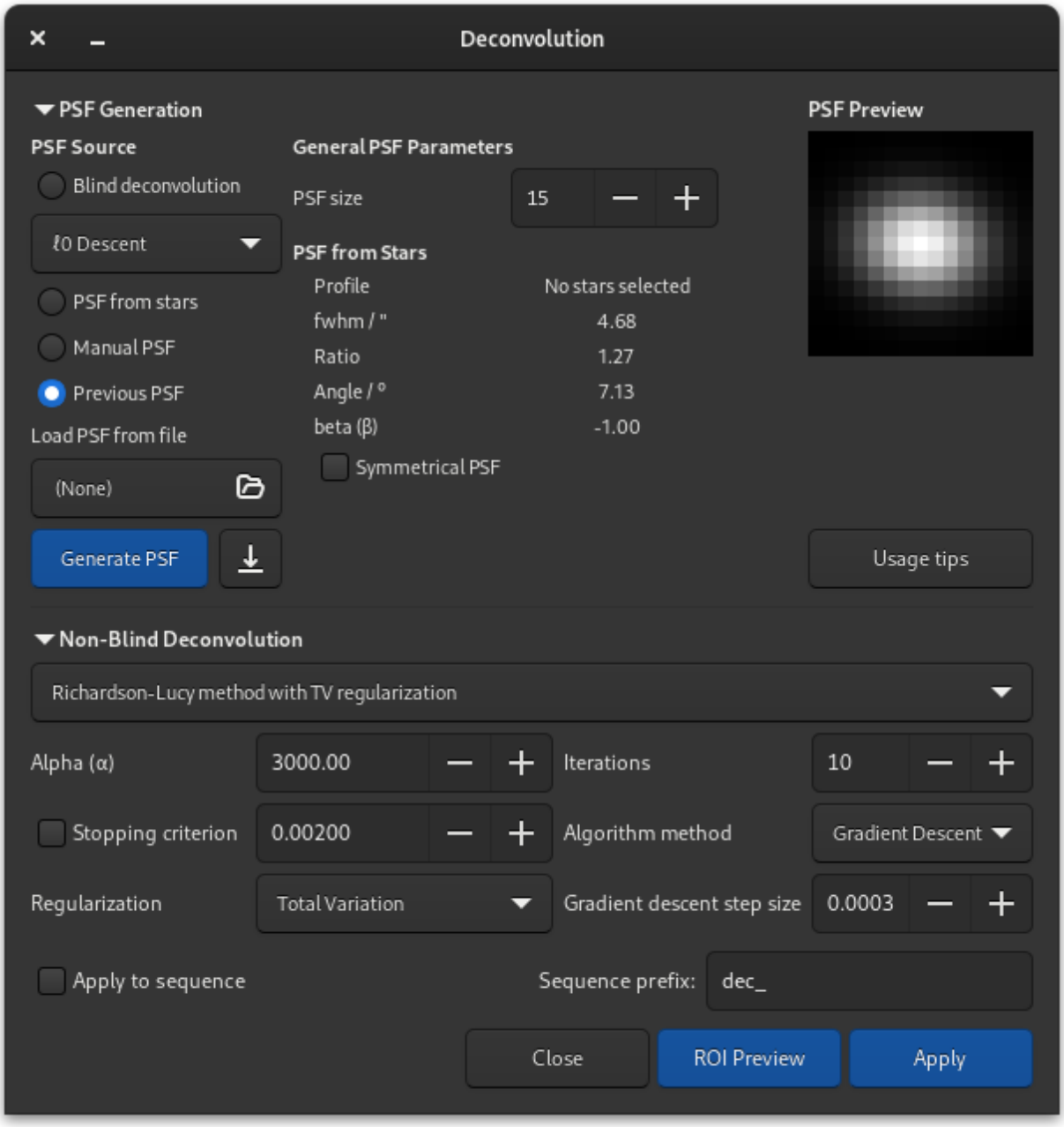


FIG. 40 – Boite de dialogue de l'outil déconvolution.

— Lors de la déconvolution d'une séquence, la PSF sera calculée uniquement pour la première image. La même PSF sera réutilisée pour toutes les images de la séquence.

Aperçu des méthodes de définition du noyau de flou

— **Descente $_0$** : Il s'agit de la méthode d'estimation de la PSF par défaut, basée sur les travaux d'Anger, Delbracio et Facciolo. Les paramètres n'ont généralement pas besoin d'être ajustés, à l'exception des PSF particulièrement grandes pour lesquelles vous pouvez essayer le modèle d'estimation des PSF à plusieurs échelles. Ce dernier est désactivé par défaut car, au cours du développement, il a été constaté qu'il avait tendance à produire des résultats peu naturels avec les tailles de PSF les plus courantes, petites ou moyennes.
— **Irrégularités spectrales** [Goldstein2012] : Cette méthode d'estimation de la PSF est proposée comme alternative. En général, elle n'est pas aussi performante que la méthode de descente $_0$, mais elle peut être utile si vous découvrez une image pour laquelle la méthode par défaut ne donne pas de bons résultats. Pour cette méthode, l'image latente nette ne doit pas contenir de bords tant que le modèle de décroissance spectrale est respecté. D'autre part, la méthode de descente $_0$ suppose un modèle similaire (puisque les bords ont la même décroissance spectrale), mais nécessite d'avoir des gradients peu denses et d'être contrastée, donc les bords doivent être en phase, donc théoriquement ce modèle peut fonctionner mieux sur des images sans étoiles et peu contrastées. Ce modèle peut donc théoriquement mieux fonctionner sur les images sans étoile à faible contraste. Il faudra probablement expérimenter pour trouver l'algorithme qui correspond le mieux à vos données.
— **PSF à partir d'étoiles** : Cette méthode modélise une PSF à partir de la PSF moyenne des étoiles sélectionnées. Il est important d'être sélectif dans le choix des étoiles : elles ne doivent pas être saturées, car cela entraînerait une distorsion importante de l'estimation de la PSF, mais elles ne doivent pas non plus être si faibles que les fonctions d'analyse d'étoiles de Siril fournissent des mesures inexactes des étoiles. Les étoiles choisies doivent être raisonnablement brillantes, assez centrales dans l'image et dans une zone de l'image avec un arrière-plan assez constant. Une fois les étoiles sélectionnées, vous pouvez choisir un modèle de profil d'étoile gaussien ou de Moffat et, lors de l'exécution de la déconvolution, la PSF sera synthétisée à partir des paramètres moyens des étoiles sélectionnées. Si aucune étoile n'est sélectionnée, Siril tentera d'auto-détecter les étoiles dont l'amplitude du pic est comprise entre 0,07 et 0,7, avec un profil de Moffat. Cette plage permet d'éviter les étoiles saturées ainsi que celles qui sont trop faibles pour donner une solution précise, et fournit généralement de bons résultats.
— **PSF manuelle** : Cette méthode vous permet de définir une PSF manuellement. Il est possible de définir des modèles de PSF gaussiens, de Moffat ou en disque. Notez que la FWHM est spécifiée en pixels et non en secondes d'arc. Les modèles gaussiens et de Moffat conviennent à la déconvolution des formes d'étoiles résultant de la distorsion atmosphérique ; le modèle de PSF de disque convient à la déconvolution de l'effet d'un léger défaut de mise au point.
— **Charger une PSF à partir d'un fichier** : Cette méthode vous permet de charger une PSF à partir de n'importe quel format d'image supporté par Siril. La PSF fournie doit être carrée (elle sera rejetée si elle n'est pas carrée) et doit être impaire (elle sera recadrée d'un pixel dans chaque direction si elle n'est pas impaire, mais cela donnera une PSF légèrement décentrée et n'est pas optimal par rapport à la fourniture d'une PSF impaire dès le début). Des PSF monochromes ou à trois canaux peuvent être chargées. Si une PSF à 3 canaux est chargée en conjonction avec une image monochrome, les valeurs de luminance uniformément pondérées de la PSF seront utilisées. Si une PSF à 3 canaux est chargée avec une image à 3 canaux, chaque canal de l'image sera déconvolué en utilisant le canal correspondant de la PSF. Si unePSF monochrome est chargée avec une image à 3 canaux, l'image sera convertie dans l'espace couleur LAB et le canal L (Luminance) sera déconvolué en utilisant la PSF monochrome pour des raisons d'efficacité de calcul, et le L déconvolué sera recombiné avec les canaux A et B et reconverti en RVB.
— **PSF précédente** : Cette méthode permet de réutiliser la PSF précédemment estimée. Elle est surtout utile avec les méthodes d'estimation de la PSF en aveugle : si vous êtes satisfait de la PSF estimée mais que vous souhaitez effectuer un certain nombre de tests en utilisant différents paramètres pour l'étape finale de la déconvolution, vous pouvez réutiliser la PSF précédente et économiser du temps de calcul.
— Une fois estimées, les PSF peuvent être sauvegardées si on le souhaite. Si Siril est compilé avec le support libtiff, la PSF sera sauvegardée au format TIFF 32 bits, avec le même nom de fichier que l'image courante, mais horodaté et suffixé par _PSF. Si Siril a été construit sans le support de libtiff, le PSF sera sauvegardé sous la forme d'un fichier FITS. Bien que ce soit le format principal de Siril pour les fichiers d'images astronomiques, le format TIFF est préféré pour les PSF : l'inconvénient d'utiliser le format FITS pour les PSF est une compatibilité réduite

avec les éditeurs d'images que vous pourriez souhaiter utiliser pour éditer ou examiner le fichier sauvegardé.

Astuce

Si la génération aveugle d'une PSF de déconvolution peut se faire sur des données linéaires et non linéaires, l'utilisation d'une PSF à partir d'une PSF d'étoile ne peut se faire que sur des images linéaires. Dans le cas contraire, les valeurs de PSF ne seraient pas valides.

Astuce

Si une ROI est défini, les méthodes d'estimation des PSF en aveugle calculeront les PSF à partir de la ROI plutôt que de l'image entière. Si vous souhaitez calculer une PSF à partir de l'image entière, vous devez effacer la ROI avant d'estimer la PSF, puis définir la ROI pour appliquer un aperçu. Cependant, vous avez également la possibilité de calculer la PSF uniquement à partir d'une partie sélectionnée de l'image. Cela peut être souhaitable si vous avez des aberrations optiques près du bord de l'image et que vous souhaitez estimer la PSF à partir de la zone centrale exclusivement.

Vue d'ensemble de la déconvolution non aveugle

— **Déconvolution Richardson-Lucy** [Lucy1974] : Il s'agit de l'algorithme de déconvolution non aveugle par défaut. Il s'agit d'une méthode itérative, célèbre pour son utilisation dans la correction des distorsions d'image au début de la période d'exploitation du télescope spatial Hubble. Dans Siril, il est régularisé en utilisant soit la méthode de variation totale, qui vise à pénaliser l'algorithme pour l'amplification du bruit, soit la norme de Frobenius de la matrice hessienne locale. Cette régularisation est basée sur les dérivées secondes. En plus de la régularisation, un paramètre d'arrêt précoce est fourni, qui permet d'arrêter l'algorithme dès que son taux de convergence tombe en dessous d'un certain niveau. L'augmentation de la valeur du paramètre d'arrêt précoce permet de réduire les artefacts autour des étoiles et des arêtes (trop) vives. Deux formulations de l'algorithme de Richardson-Lucy sont fournies : la formulation multiplicative et la formulation de descente de gradient. Cette dernière permet un meilleur contrôle, car la taille du pas de descente du gradient peut être modifiée (l'inconvénient est qu'en utilisant des plus petits pas, plus d'itérations sont nécessaires pour atteindre le même niveau de convergence). Le plus grand avantage de la méthode de descente de gradient est qu'elle permet d'utiliser une plus grande régularisation - ce qui peut être problématique dans l'algorithme multiplicatif de Richardson-Lucy car le terme de régularisation apparaît dans le dénominateur et de petites valeurs ici (forte régularisation) peuvent causer de l'instabilité. Siril utilisera la convolution naïve pour les noyaux de petite taille et la convolution basée sur les FFT pour les noyaux de plus grande taille où les FFT fournissent un algorithme plus efficace. (Ceci est automatique et ne nécessite aucune intervention de l'utilisateur.)
— **Méthode de filtrage de Wiener** : Cette méthode est une méthode de déconvolution non itérative. Elle modélise un profil de bruit supposé gaussien, c'est-à-dire un bruit modélisé par un profil constant. La constante alpha est utilisée pour définir la force de régularisation par rapport au niveau de bruit. Comme pour les autres algorithmes, une valeur plus petite pour alpha permet une plus grande régularisation. Cet algorithme peut être bon pour les images lunaires où le régime de bruit est gaussien et non de Poisson, mais il fonctionne généralement mal sur les images de ciel profond où le bruit a toujours tendance à avoir un caractère de Poisson.
— **Méthode de Split Bregman** : Cette méthode est utilisée en interne dans les processus d'estimation de la PSF et est également proposée comme algorithme de déconvolution en phase finale. Il s'agit d'un algorithme couramment utilisé pour résoudre les problèmes d'optimisation convexe. Cet algorithme est également régularisé à l'aide d'une fonction de coût de variation totale. Il n'est pas aussi performant que l'algorithme de Richardson-Lucy pour les paysages étoilés, mais il peut être envisagé pour les images sans étoiles ou les images de la surface lunaire.

Astuce

Le choix de la méthode de déconvolution est très important pour obtenir de bons résultats. En général, pour les images de ciel profond, il est important d'utiliser une méthode de Richardson-Lucy : les méthodes de Split Bregman et de Wiener donnent de mauvais résultats autour des étoiles en raison de la gamme dynamique extrême. Pour les images linéaires, il est généralement préférable d'utiliser les méthodes de Richardson-Lucy à descente de gradient, et si des artefacts apparaissent autour des étoiles brillantes, il faut alors réduire la taille du pas. Cette approche réduit l'impact de chaque itération, ce qui nécessite un plus grand nombre d'itérations, mais cela signifie que vous pouvez obtenir un contrôle plus fin en effectuant la déconvolution jusqu'au point où les artefacts commencent à se former, puis en revenant très légèrement en arrière. Pour les images étirées, les algorithmes multiplicatifs de Richardson-Lucy peuvent être utilisés.

Astuce

Pour les images lunaires et planétaires empilées, les méthodes de Split Bregman ou de Wiener peuvent être plus appropriées. Ces méthodes ne nécessitent généralement pas d'itération comme le fait Richardson-Lucy, et elles peuvent être mieux adaptées aux caractéristiques de bruit des images empilées à rapport signal/bruit élevé. (L'algorithme de Richardson-Lucy est basé sur l'hypothèse d'un bruit de Poisson, ce qui est généralement vrai pour l'imagerie de ciel profond, alors que la méthode de Wiener mise en œuvre ici suppose une distribution de bruit gaussienne qui peut mieux convenir aux images planétaires/lunaires empilées).

Paramètres et Réglages

Paramètres généraux

- Taille PSF. La taille de la PSF d'entrée doit être choisie suffisamment grande pour garantir que la PSF est incluse dans le domaine spécifié. Toutefois, si elle est trop grande, les méthodes d'estimation aveugle de la PSF peuvent donner des résultats moins bons et plus longs à calculer.
- Lambda (λ). Paramètre de régularisation pour l'estimation de la PSF. Essayez de diminuer cette valeur pour les images bruitées.

Paramètres d'estimation de la PSF de descente de $_0$

- Multi-échelle. Ce paramètre active l'estimation de la PSF à plusieurs échelles. Cela peut aider à stabiliser l'estimation de la PSF lorsque l'on spécifie une grande taille de PSF, mais certaines PSF générées avec cette option peuvent donner lieu à des résultats peu naturels et elle est donc désactivée par défaut.
- Paramètres experts. Ils ne devraient normalement pas nécessiter de réglage, mais sont mis à la disposition des curieux.
  - Gamma définit la force de régularisation utilisée lors de l'étape de prédiction de l'image déconvoluée. Pour un gamma donné, l'estimation devient aussi plus bruitée à mesure que le bruit augmente. Si gamma est augmenté, l'estimation est moins affectée par le bruit mais tend à être plus lisse. La valeur par défaut de 20 a été déterminée expérimentalement dans [Anger2019].
  - Itérations définit le nombre d'itérations utilisées dans la procédure d'estimation de la PSF. Les auteurs de l'algorithme indiquent qu'il n'y a que peu d'avantages à augmenter ce nombre à 3 et qu'il n'y a aucun avantage à l'augmenter au-delà de 3.
  - Le rapport lambda et le minimum lambda définissent les paramètres permettant d'affiner la prédiction de l'image nette par le biais de valeurs successives du paramètre de régularisation du prédicteur d'image nette à chaque itération de la méthode.
  - Le facteur d'échelle, le flou de suréchantillonnage et Sous-échantillonner le flou ne sont utilisés que lorsque l'estimation multi-échelle est active. Ils définissent le facteur d'échelle par défaut entre chaque niveau d'échelle et la quantité de flou à utiliser lors de la remise à l'échelle entre chaque échelle.
  - Seuil du noyau. Les valeurs inférieures à ce niveau dans l'estimation de la PSF sont mises à zéro.

Paramètre de la détermination de la PSF par irrégularité spectrale

— Le facteur de compensation contrôle la force d'un filtre utilisé pour éviter une netteté excessive dans la PSF estimée. Pour les images présentant un flou intrinsèque, il convient d'utiliser une valeur proche de l'unité. Pour les images intrinsèquement nettes, des valeurs faibles peuvent entraîner des artefacts et la valeur doit être augmentée jusqu'à un grand nombre, ce qui désactive effectivement le filtre.
— Paramètres experts. Ils ne devraient normalement pas nécessiter de réglage, mais sont mis à la disposition des curieux.
  — Itérations de la boucle interne définit le nombre d'itérations effectuées dans la boucle intérieure de la méthode d'irrégularité spectrale. L'algorithme converge rapidement et il peut être possible de réduire ce nombre à environ 100 sans trop dégrader le résultat.
  — Échantillons par boucle externe. Ce paramètre contrôle le nombre de phases aléatoires à échantillonner. Étant donné que la récupération de la phase commence par des valeurs aléatoires pour chaque échantillon, il est important de tirer suffisamment d'échantillons pour éviter de converger vers un minimum local. La PSF se stabilise rapidement pour les images à faible bruit, mais si vous cherchez à améliorer les résultats de cette méthode, c'est le premier des paramètres experts à essayer d'ajuster, en particulier avec des images présentant des niveaux de bruit plus élevés.
  — Itération de la boucle externe. [Anger2018], suggère que 2 itérations peuvent suffire à produire une estimation plausible de la PSF, et qu'il est négligeable d'augmenter ce nombre au-delà de 3.

PSF des étoiles

— Cette méthode de génération de PSF n'a pas de paramètres ajustables. Elle génère une PSF basée sur les paramètres moyens des étoiles sélectionnées, en utilisant la commande *findstar* ou la boîte de dialogue *PSF Dynamique*. Les paramètres moyens sont affichés dans la boîte de dialogue de déconvolution lorsque cette méthode de génération de PSF est sélectionnée. Il est préférable que l'utilisateur sélectionne activement les étoiles qu'il souhaite utiliser pour cette méthode, afin d'obtenir la PSF la plus précise. Idéalement, une dizaine d'étoiles assez brillantes mais non saturées devraient être sélectionnées dans la région centrale de l'image (pour exclure les étoiles qui peuvent souffrir de coma ou d'autres aberrations). Cependant, si l'utilisateur n'a sélectionné aucune étoile, Siril tentera de détecter automatiquement les étoiles appropriées en exécutant sa routine de détection avec des filtres réglés pour ne conserver que les étoiles dont les amplitudes de crête sont comprises entre 0,07 et 0,7. Cette plage permet d'éviter les étoiles saturées et celles qui sont trop faibles pour donner une solution précise. Cette méthode fonctionne bien dans la plupart des cas, mais peut encore être affectée par des aberrations hors centre.
— Si vous cochez la case *PSF symétrique*, la PSF générée sera circulaire. Cela correspondra à la moyenne des FWHM et beta des étoiles sélectionnées mais ne correspondra à aucune élongation.

PSF manuel

Cette méthode de génération de PSF permet de spécifier une PSF paramétrique personnalisée.
— Profil permet de choisir le profil de la PSF. Les profils Gaussien, Moffat, disque et dique d'Airy sont supportés.
  — Les PSF gaussiennes et de Moffat sont utilisées pour faire correspondre les paramètres des étoiles mesurés à partir de l'image. Elles devraient fournir une bonne estimation de la fonction de flou totale appliquée à l'image, car les étoiles sont des sources ponctuelles.

Fig. 41 – Un exemple d'une PSF avec un profil Moffat avec fwhm=5", Angle=45°, Ratio=1.20, $\beta = 4.5$ et une taille de PSF de 15.

  — Les PSF disque sont utilisés pour déconvoluer les images qui ne sont pas au point.
  — La PSF de disque d'Airy est utilisée pour déconvoluer la diffraction qui est la conséquence physique de la diffraction par l'ouverture de votre télescope.

FIG. 42 – Un exemple de profil Disque avec fwhm=5" et une taille de PSF de 15.

FIG. 43 – Exemple d'une PSF d'un disque d'Airy avec Diamètre = 250mm, Longueur focal = 4500mm, Longueur d'onde = 525nm, Taille Pixel = 2.9µm, Obstruction Centrale = 40% et avec une taille de PSF de 41.

- FWHM spécifie la largeur à mi-maximum du profil choisi (pour les PSFs de disque, il spécifie simplement le rayon).
- Beta (β) spécifie le paramètre beta utilisé pour le profil de PSF Moffat. Il est ignoré par les autres profils de PSF.
- Pour les PFS du disque d'Airy, un certain nombre de paramètres de votre télescope et de votre capteur sont nécessaires :
  - Ouverture
  - Longueur focale
  - Taille de pixel du capteur
  - Longueur d'onde centrale de l'image. Siril essaiera d'extraire ces données des métadonnées de l'image lorsqu'elles sont disponibles, mais si certains paramètres sont manquants ou semblent déraisonnables, Siril les mettra en évidence et imprimera un avertissement dans le log vous recommandant de les vérifier. Le rapport de l'obstruction centrale est également nécessaire pour générer un disque d'Airy précis. Il est exprimé en pourcentage, c'est-à-dire que la surface totale de l'obstruction centrale est divisée par la surface totale de l'ouverture x 100. Pour les réfracteurs (avec des lentilles), ce pourcentage est nul ; pour les autres télescopes, il varie : il peut être d'environ 20 % pour un réflecteur newtonien ou de 40 à 50 % pour certains télescopes Dall-Kirkham corrigés. Vous devrez mesurer votre instrument ou consulter les spécifications du fabricant.

Déconvolution de Richardson-Lucy

Les paramètres utilisés pour configurer la déconvolution de Richardson-Lucy dans Siril sont les suivants :

- **alpha** définit l'intensité de la régularisation. Une valeur plus petite d'alpha donne une régularisation plus forte et un résultat plus lisse ; une valeur plus grande réduit la force de régularisation et préserve plus de détails de l'image, mais peut entraîner l'amplification du bruit.
- **Itérations** spécifie le nombre maximum d'itérations à utiliser. En l'absence de bruit, un grand nombre d'itérations fera converger l'estimation de la déconvolution plus près de l'image réelle, mais un nombre excessif d'itérations amplifiera également le bruit et provoquera des artefacts en annaux autour des étoiles. La valeur par défaut est 1 itération : un nombre plus élevé peut être défini pour calculer plusieurs itérations automatiquement, ou vous pouvez continuer à appuyer sur *Appliquer* pour appliquer une itération à la fois jusqu'à ce que vous soyez satisfait du résultat. (Ou aller plus loin, décider que vous n'êtes plus satisfait et utiliser *Annuler*.)
- **Critère d'arrêt** définit un critère de convergence basé sur les différences d'estimation successives. L'algorithme s'arrête lorsque la convergence se situe dans la limite spécifiée. Il s'agit d'un paramètre important - si vous obtenez des anneaux autour des étoiles dans votre image finale, essayez d'augmenter la valeur du critère d'arrêt. Ce paramètre peut être complètement désactivé à l'aide du bouton à cocher.
- **Méthode d'algorithme** spécifie s'il faut utiliser l'implémentation multiplicative ou l'implémentation par descente de gradient.
- **Taille du pas** spécifie la taille du pas à utiliser pour l'implémentation de la descente de gradient. Ce paramètre ne doit pas être trop grand, sinon l'algorithme ne convergera pas. Ce paramètre n'a aucun effet si l'implémentation multiplicative est sélectionnée.

> **Astuce**
>
> Pour les images linéaires, l'utilisation des méthodes de descente de gradient permet d'obtenir le contrôle nécessaire pour éviter la formation d'anneaux autour des étoiles. Cependant, pour la déconvolution d'images étirées, cette méthode peut être inutilement lente. L'utilisation de méthodes multiplicatives permet donc souvent de gagner du temps sans compromettre la qualité de l'image.

Déconvolution de Split Bregman

Les paramètres utilisés pour configurer la déconvolution de Split Bregman dans Siril sont les suivants :

- **alpha** définit l'intensité de la régularisation. Une valeur plus petite d'alpha donne une régularisation plus forte et un résultat plus lisse ; une valeur plus grande réduit la force de régularisation et préserve plus de détails de l'image, mais peut entraîner l'amplification du bruit.
- **Itérations** spécifie le nombre maximum d'itérations à utiliser. La méthode Split Bregman ne nécessite pas d'itérations multiples dans la forme implémentée ici, mais peut être itérée si on le souhaite. Cela ne fait généralement qu'une petite différence et la valeur par défaut est donc 1.

Déconvolution de Wiener

La déconvolution de Wiener dans Siril ne nécessite qu'un seul paramètre :

- **alpha** définit l'intensité de la régularisation. Une valeur plus petite d'alpha donne une régularisation plus forte et un résultat plus lisse ; une valeur plus grande réduit la force de régularisation et préserve plus de détails de l'image, mais peut entraîner l'amplification du bruit.

Paramètres de performance de la FFTW

Les algorithmes d'estimation et de déconvolution de la PSF font un usage intensif des transformées de Fourier rapides à l'aide de la bibliothèque FFTW. Ceci offre un certain nombre d'options de réglage, qui peuvent être ajustées dans l'onglet performance du dialogue principal Siril *préférences*.

Note sur l'ordre des lignes de l'image

Les différents types d'images traitées par Siril peuvent avoir leurs données de pixels arrangées dans des ordres différents. Les fichiers vidéo SER stockent toujours les données de haut en bas, tandis que les fichiers FITS peuvent stocker les données de bas en haut ou de haut en bas. L'ordre ascendant est la recommandation initiale, mais de plus en plus de fichiers FITS proviennent de caméras CMOS qui ont tendance à suivre un ordre descendant des pixels.

Lorsqu'une image est déconvoluée avec une PSF créée à partir de la même image (ou ouverte), cela ne pose aucun problème. Cependant, des problèmes peuvent survenir si une PSF est générée à partir d'une image avec un ordre des lignes et utilisée pour déconvoluer une image ou une séquence avec l'ordre des lignes opposé. Il s'agit d'un cas d'utilisation particulier, mais son traitement systématique donne lieu à un comportement qui, à première vue, peut être surprenant : il est donc expliqué ci-dessous.

Siril traite le problème en suivant l'ordre des lignes de l'image avec laquelle le PSF a été créé. Les PSF sont toujours sauvegardés en utilisant un ordre de rangée ascendant (en les retournant automatiquement s'ils ont été créés avec une image descendante), et lorsqu'ils sont chargés, l'ordre de rangée correspond à l'ordre de rangée de l'image actuellement ouverte. Si une image dont l'ordre des lignes est opposé est ouverte, l'ordre des lignes de la PSF sera modifié pour correspondre. Cela signifie que si, par exemple, vous prenez des images FITS ascendantes, utilisez l'une d'entre elles pour générer une PSF, puis les convertissez en une séquence SER descendante, la PSF sera convertie dans l'orientation correcte pour correspondre à la séquence SER. Si une PSF est en cours de prévisualisation au moment où une image avec l'ordre des lignes opposé est ouverte, la prévisualisation ne sera pas mise à jour immédiatement : le changement d'ordre des lignes sera détecté automatiquement et la PSF sera inversée au moment où elle sera appliquée à l'image.

Galerie

Cette section présente quelques exemples de cas où la déconvolution a mal tourné, ainsi que des explications sur les raisons de ces erreurs.

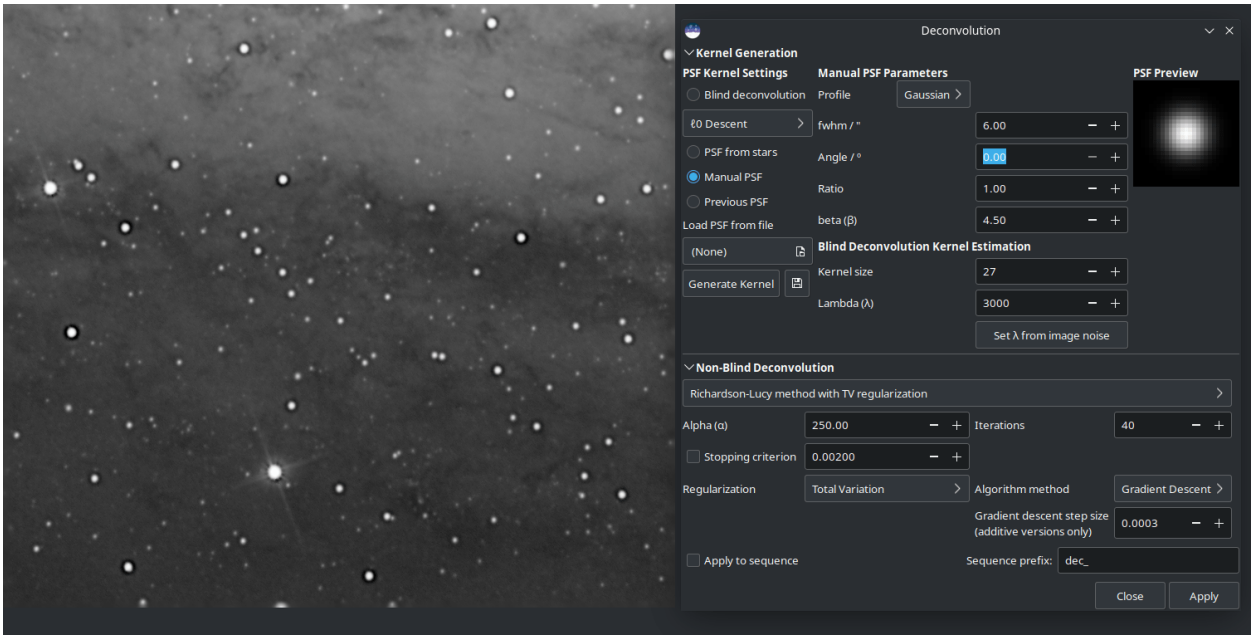


Fig. 44 – La PSF spécifiée manuellement était trop grande, ce qui se traduisait par de grands anneaux sombres autour des étoiles.

Déconvolution : Conseils d'utilisation

Vous êtes arrivé ici en cliquant sur le bouton "Conseils" de l'outil de déconvolution dans Siril. Ne vous inquiétez pas : la déconvolution est une technique délicate. Même en théorie, elle est vraiment difficile : il n'y a aucune garantie que les mathématiques convergent toujours vers une solution unique qui améliore votre image. Cela dit, voici quelques conseils pour vous aider à tirer le meilleur parti des algorithmes de déconvolution de Siril.

Quelle PSF utiliser ?

L'utilisation d'une PSF précise est fondamentale pour obtenir de bons résultats avec la déconvolution. Les deux façons les plus simples de générer une PSF sont d'utiliser une estimation aveugle de la PSF, ou de modéliser votre PSF sur les étoiles de votre image.

PSF des étoiles

Siril peut détecter et modéliser les étoiles dans votre image. Voir la page d'aide de Dynamic PSF pour plus de détails. Pour obtenir un bon modèle pour votre PSF, essayez de sélectionner le profil d'étoile de Moffat dans PSF Dynamique. Les étoiles sont des sources ponctuelles, donc la fonction d'étalement d'une étoile moyenne est un bon modèle pour les effets de flou que nous essayons d'éliminer par déconvolution.

> **Astuce**
>
> Une fois les étoiles détectées, triez-les par amplitude de pic (paramètre "A"). Sélectionnez et supprimez celles dont l'amplitude est supérieure à 0,7 ou inférieure à 0,1, et si votre image contient des galaxies de fond, vérifiez qu'il ne

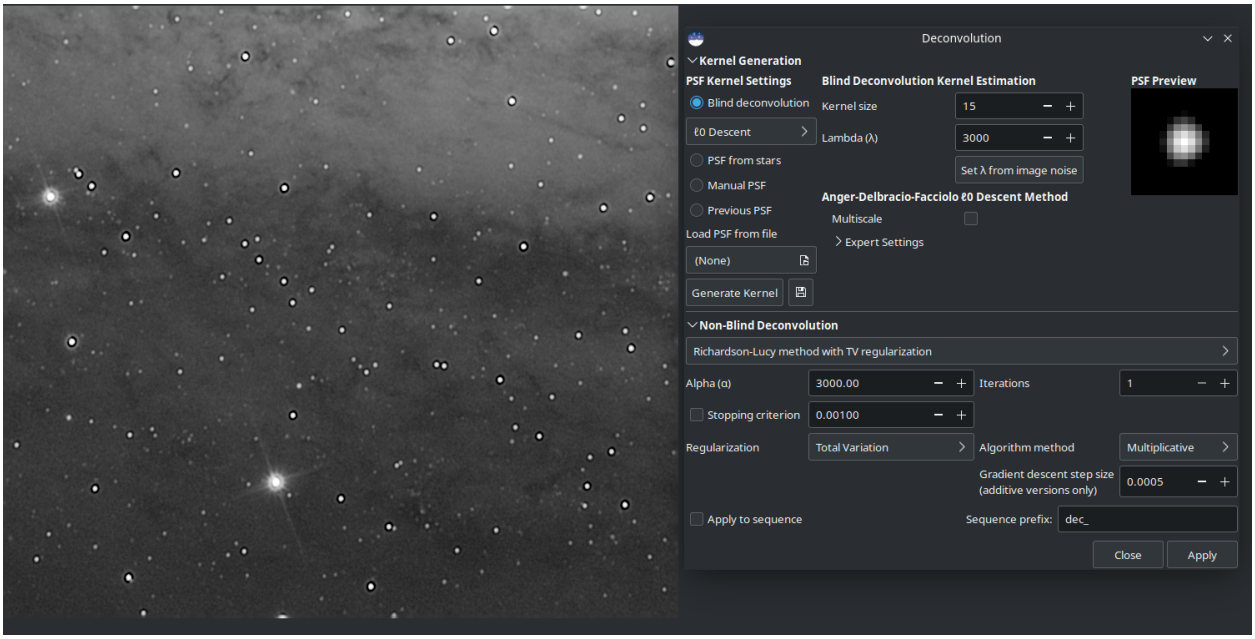


FIG. 45 – Trop d'itérations ont été appliquées. (Je les ai appliquées une par une pour exagérer le résultat, c'est pourquoi le paramètre itérations indique toujours 1.)

reste pas de faux positifs. Les étoiles dans cette gamme de luminosité ne sont pas saturées et pas trop faibles pour donner un modèle PSF précis.

Astuce

Si la génération aveugle d'une PSF de déconvolution peut se faire sur des données linéaires et non linéaires, l'utilisation d'une PSF à partir d'une PSF d'étoile ne peut se faire que sur des images linéaires. Dans le cas contraire, les valeurs de PSF ne seraient pas valides.

Estimation de la PSF en aveugle

Ces méthodes peuvent estimer automatiquement une PSF sur la base de l'image elle-même. Si vous n'avez pas de meilleures connaissances préalables de la PSF, comme des étoiles dans l'image (par exemple, une imagerie lunaire qui ne contient pas d'étoiles), cela peut être votre meilleure option. Dans la plupart des cas, il est recommandé d'utiliser la méthode par défaut $_0$: elle est plus rapide et donne généralement de meilleurs résultats.

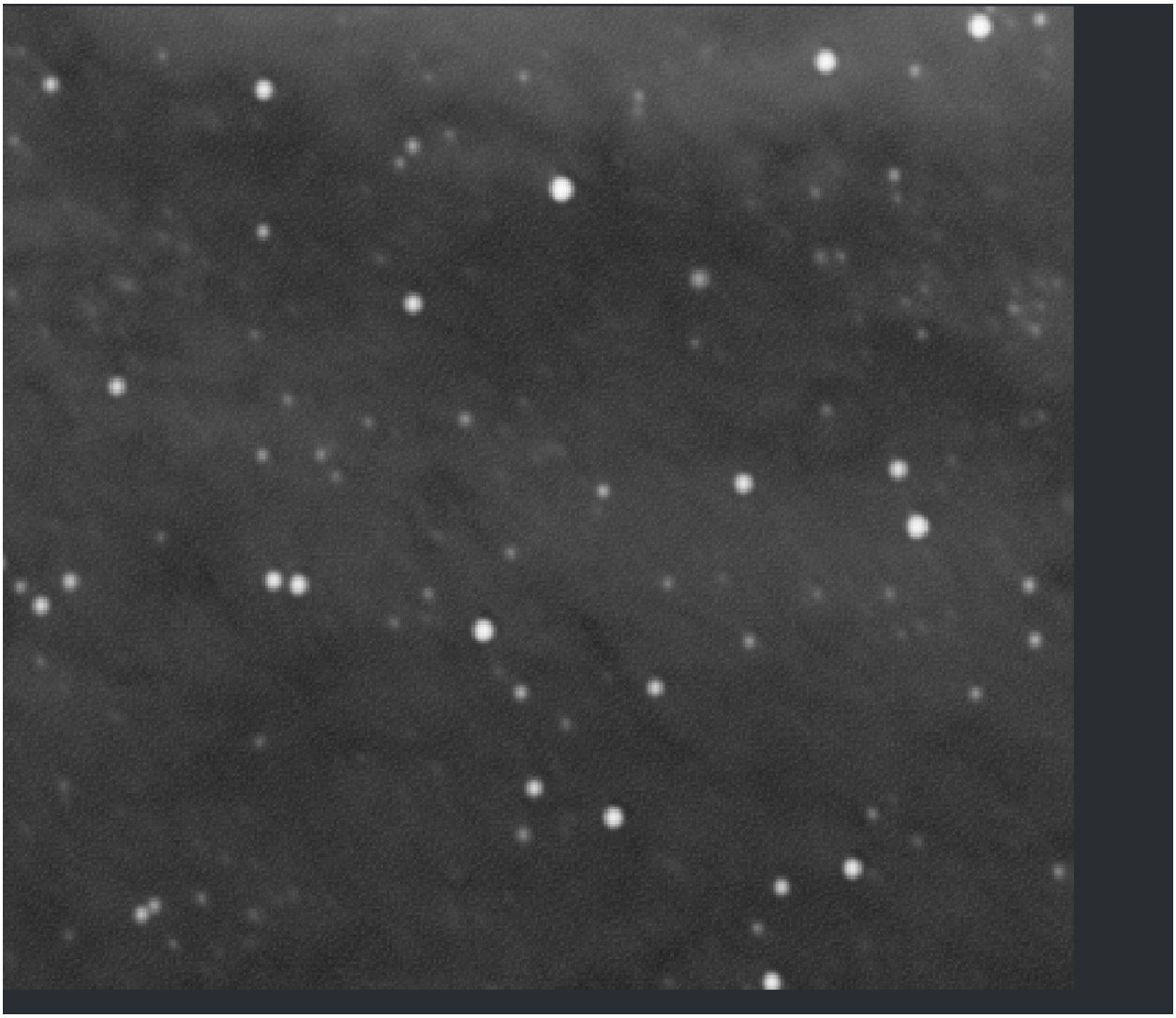


FIG. 46 – Gros plan montrant l'effet d'une tentative d'application d'une régularisation trop importante (alpha = 30) à l'aide de la version multiplicative de Richardson-Lucy. Pour une forte régularisation et/ou un meilleur contrôle de chaque itération, la formule de descente de gradient est recommandée.

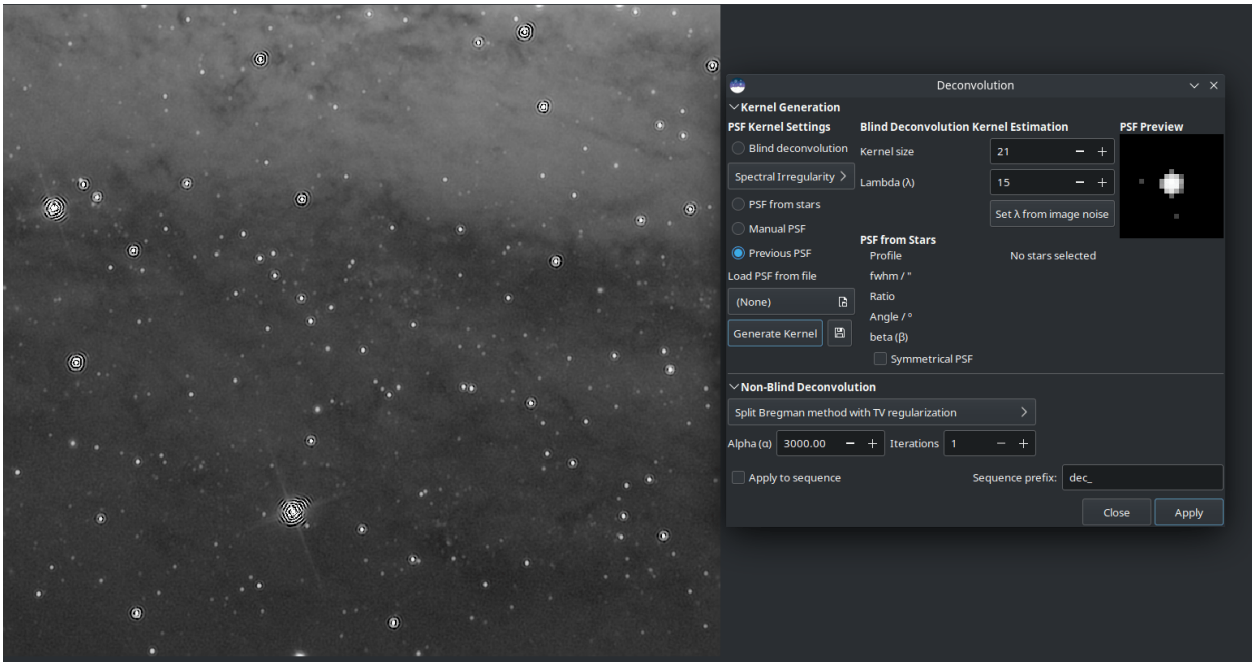


Fig. 47 – Exemple typique de tentative de déconvolution d'un champ d'étoiles d'une image non étirée à l'aide de la méthode de Split Bregman (dans ce cas) ou de la déconvolution par filtre de Wiener. Ces méthodes sont plus adaptées aux images planétaires, lunaires ou solaires ; pour les paysages stellaires, la méthode Richardson-Lucy est toujours recommandée.

Astuce

Quelle que soit la façon dont vous générez votre PSF, vérifiez l'aperçu pour vous assurer qu'elle ne semble pas tronquée. Si c'est le cas, augmentez la taille de la PSF jusqu'à ce qu'aucune partie significative de la PSF ne soit coupée.

Autres méthodes de génération de PSF

D'autres méthodes de génération de PSF méritent d'être mentionnées : le profil de disque manuel et le disque d'Airy. Le profil de disque peut être utilisé pour améliorer les images dont la mise au point est légèrement décalée. Essayez de faire correspondre la taille du disque à la taille du flou hors foyer. Le disque d'Airy peut être utilisé pour corriger le léger flou causé par la diffraction du tube du télescope lui-même.

Astuce

Si vous avez un seeing exceptionnel (peu ou pas de flou atmosphérique), la déconvolution de l'image à l'aide d'un disque d'Airy peut suffire.

La déconvolution de l'image

Une fois que vous avez généré une PSF dont vous êtes satisfait, vous êtes prêt à faire une déconvolution de votre image. Il est important d'utiliser les bons paramètres pour obtenir de bons résultats.

Astuce

Deconvolution is quite slow for large images. To make it quicker to find the best parameters, save your work at this point and use the ROI feature.

Images avec des étoiles

Les images contenant des étoiles, en particulier les données linéaires (non étirées), doivent toujours être déconvoluées à l'aide des méthodes de Richardson-Lucy. Ignorez Split Bregman et Wiener : ces algorithmes sont adaptés aux images d'objets du système solaire.

Les images du ciel profond posent deux défis à la déconvolution : l'anneau autour des étoiles brillantes et l'amplification du bruit de fond.

Pour traiter les anneaux autour des étoiles, essayez d'utiliser la méthode de descente de gradient et augmentez progressivement le nombre d'itérations jusqu'à ce que vous commenciez à voir des signes d'anneaux sombres se former autour des étoiles, puis réduisez légèrement les itérations.

L'animation ci-dessus montre l'effet de la réduction du nombre d'itérations de la formulation multiplicative de Richardson-Lucy : elle démontre également le contrôle plus fin qui peut être obtenu en utilisant la méthode de descente de gradient, au prix d'un plus grand nombre d'itérations.

Pour traiter l'amplification du bruit de fond, vous pouvez essayer d'appliquer une petite réduction du bruit avant la déconvolution. Dans la boîte de dialogue Réduction de bruit choisissez l'algorithme de débruitage secondaire Anscombe VST et laissez la modulation assez basse, essayez autour de 50-60%. Vous voulez juste atténuer le bruit pour vous permettre de pousser le nombre d'itérations un peu plus loin, sans générer une image complètement lisse.

Images Lunaires

En général, vous pouvez souhaiter augmenter la netteté d'une image lunaire après l'avoir empilée. Les images lunaires empilées peuvent être très bien affinées en utilisant les méthodes Split Bregman ou Wiener. Je choisis habituellement la méthode Split Bregman. Essayez de laisser la valeur de α par défaut, et faire la déconvolution de l'image en utilisant une PSF $_0$ estimée en aveugle. Un exemple de ceci est montré ci-dessous en utilisant une image lunaire fraîchement empilée (c'est-à-dire qu'aucun traitement par ondelettes n'a été effectué). Malgré les limitations du format d'animation GIF, l'accentuation est clairement visible ; il est également clair que les résultats de Split Bregman et Wiener sont très similaires.

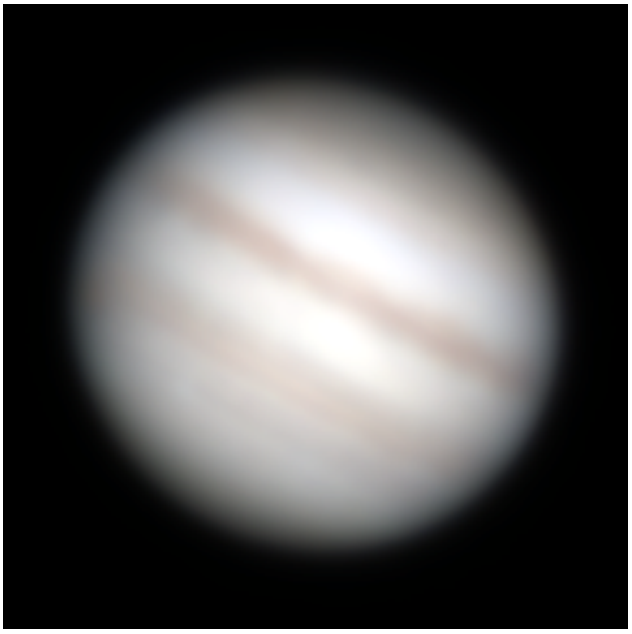
Images planétaires empilées

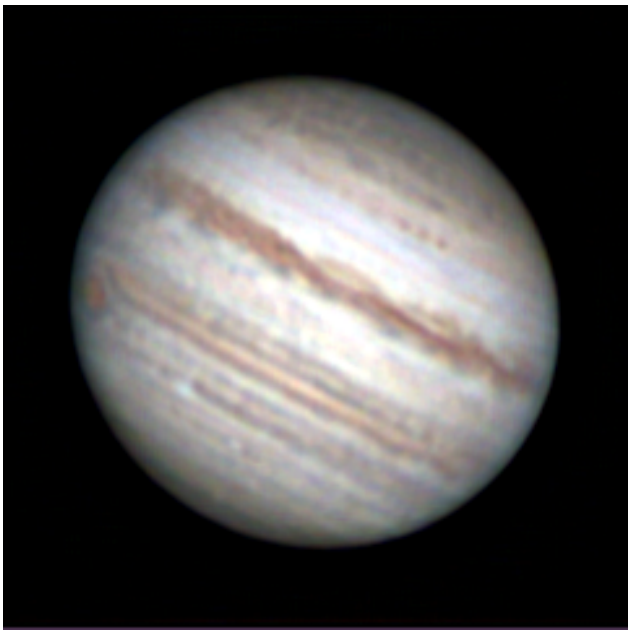
Un flux de travail planétaire typique implique l'empilement de la vidéo SER planétaire dans un outil spécialisé tel que Autostakkert ! ou Astrosurface, puis l'accentuation de l'image résultante à l'aide d'ondelettes et de déconvolution. Une combinaison de l'outil Siril A trous Wavelets et de l'outil Deconvolution donne d'excellents résultats, comme illustré ici. Cette image de Jupiter a été initialement rendue plus nette à l'aide d'ondelettes, le premier contrôle de couche étant réglé sur 75, le second sur 10 et les autres sur la valeur par défaut. Une PSF couleur a ensuite été construite à partir de 3 disques d'Airy calculés pour le télescope et le capteur utilisés (un Newtonian 6" avec une lentille de Barlow 3x et un capteur ASI462MC avec des pixels de 2,9 microns) et composée à l'aide de l'outil de composition RVB. Ceci a été utilisé pour faire une déconvolution de l'image avec 6 itérations de Richardson-Lucy (ici j'ai utilisé la version multiplicative). A chaque étape, l'image devient plus nette.

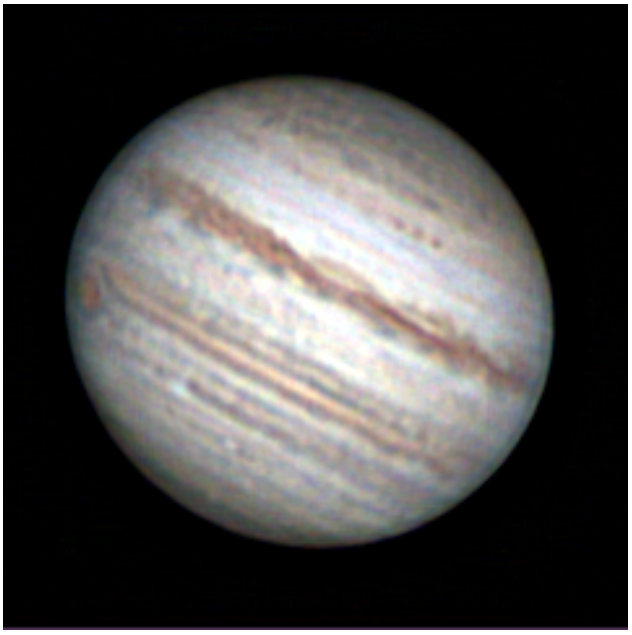
Empilement brute, toujours floue.

Traitement avec décomposition en ondelettes de Siril, force de la couche d'ondelettes 1, 75, force de la couche d'ondelettes 2, 10.

Traitée avec les ondelettes de Siril comme ci-dessus, puis avec 6 itérations de déconvolution multiplicative de Richardson-Lucy.





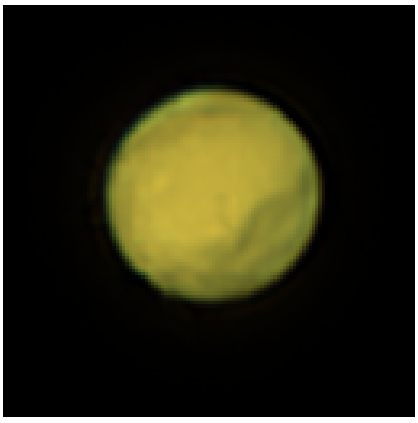


Les séquences planétaires non empilées

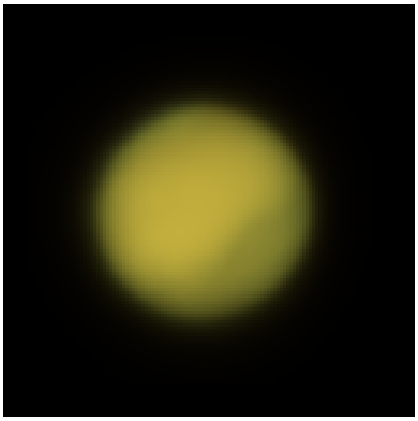
Astuce

Attention : cette méthode est extrêmement lente car elle nécessite le traitement individuel de 30 000 images (ou plus) dans une séquence planétaire !

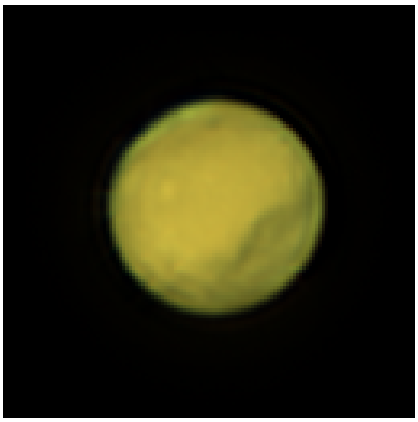
Certains utilisateurs ont suggéré d'atténuer la diffraction du télescope en faisant une déconvolution de la séquence à l'aide d'une PSF de disque d'Airy. Pour ce faire, avec une caméra planétaire couleur typique, la séquence doit être dématricée à la volée. Vous pouvez aller plus loin si vous le souhaitez en générant trois disques d'Airy séparés pour les longueurs d'onde rouge, verte et bleue (typiquement 600nnm, 530nm et 450nm respectivement). Siril ne peut pas générer directement un PSF couleur (l'interface utilisateur de la déconvolution est suffisamment complexe !) mais si vous enregistrez séparément chacun des disques d'Airy rouge, vert et bleu, vous pouvez les combiner en une PSF couleur en utilisant l'outil de composition RVB. Sauvegardez-le, et si une couleur ou une séquence est chargée, la PSF se chargera en couleur et procédera à la déconvolution de chaque canal de couleur en utilisant la PSF appropriée.



Empilés et netteté renforcée sans déconvolution individuelle des images.



Pile brute : les meilleurs 30% de 91k images déconvoluées individuellement en utilisant Siril.



Résultat de l'accentuation de la pile ou une déconvolution a été appliquée sur chaque image individuellement.

Dans l'image ci-dessus, une légère amélioration de la forme du bord est évidente dans la version qui a subit une déconvolution image par image avec une PSF de disque d'Airy en utilisant la méthode Richardson-Lucy de Siril avant l'empilement, mais il faut faire attention pour éviter la perte de détails. Ce processus est très lent : ma machine de développement a mis 4,5 heures pour effectuer la déconvolution de chacune des 91 000 images de cette séquence, et l'amélioration peut être mineure, voire nulle.

Commandes

Ligne de commande Siril

```
makepsf clear
makepsf load filename
makepsf save [filename]
makepsf blind [-l0] [-si] [-multiscale] [-lambda=] [-comp=] [-ks=] [-savepsf=]
makepsf stars [-sym] [-ks=] [-savepsf=]
makepsf manual { -gaussian | -moffat | -disc | -airy } [-fwhm=] [-angle=] [-ratio=] [-
↪beta=] [-dia=] [-fl=] [-wl=] [-pixelsize=] [-obstruct=] [-ks=] [-savepsf=]
```

Génère une PSF à utiliser avec la déconvolution, l'une des trois méthodes exposées par les commandes RL, SB ou WIENER. L'un des éléments suivants doit être fourni comme premier argument : **clear** (efface la PSF existante),

load (charge une PSF à partir d'un fichier), **save** (sauvegarde la PSF actuelle), **blind** (estimation aveugle de la PSF), **stars** (génère une PSF basée sur les étoiles mesurées dans l'image) ou **manual** (génère une PSF manuellement basée sur une fonction et des paramètres).

Aucun argument additionnel n'est requis quand l'argument **clear** est utilisé.

Pour charger une PSF précédemment sauvegardée, l'argument **load** nécessite le nom du fichier *filename* de la PSF comme second argument. Ce fichier peut être dans n'importe quel format pris en charge par Siril, mais l'image doit être au format carré et idéalement impair.

Pour sauvegarder une PSF précédemment générée, l'argument **save** est utilisé. Un nom de fichier peut être fourni en option (il doit avoir l'une des extensions ".fit", ".fits", ".fts" ou ".tif"), mais si aucun nom n'est fourni, la PSF sera nommée en fonction du nom du fichier ou de la séquence ouvert(e).

Pour l'argument **blind**, les arguments optionnels suivants peuvent être fournis : **-l0** utilise la méthode de descente l0, **-si** utilise la méthode d'irrégularité spectrale, **-multiscale** configure la méthode l0 pour effectuer une estimation PSF multi-échelle, **-lambda=** fournit la constante de régularisation.

Pour les PSF des **étoiles** détectées, le seul paramètre optionnel est **-sym**, qui configure la PSF pour qu'elle soit symétrique.

Pour une PSF **manuelle**, un des arguments **-gaussian**, **-moffat**, **-disc** or **-airy** peut être fourni afin de spécifié la fonction PSF. Pour la PSF Gaussienne ou Moffat les arguments optionels **-fwhm=**, **-angle=** and **-ratio=** peuvent être ajoutés. Pour la PSF Moffat l'argument optionel **-beta=** peut aussi être ajouté. Si ces valeurs sont omises, elles prennent par défaut les mêmes valeurs que dans fenêtre déconvolution. Pour les PSF "disc" seul l'argument **-fwhm=** est nécessaire, qui, pour cette fonction est utilisé pour donner le *diamètre* de la PSF. Pour les PSF "airy" les arguments suivant peuvent être ajoutés : **-dia=** (définit le diamètre du télescope), **-fl=** (définit la longueur focale du télescope), **-wl=** (définit la longueur d'onde pour calculer la tache d'Airy), **-pixelsize=** (définit la taille des pixels du capteur), **-obstruct=** (définit le l'obstruction centrale comme pourcentage de la surface totale d'ouverture). Si ces paramètres ne sont pas fournis, la longueur d'onde sera par défaut de 525nm et l'obstruction centrale sera par défaut de 0%. Siril tentera de lire les autres paramètres de l'image ouverte, mais certains logiciels d'imagerie peuvent ne pas les fournir tous, auquel cas vous obtiendrez de mauvais résultats, et notez que les métadonnées peuvent ne pas être remplies pour les vidéos au format SER. Vous apprendrez par l'expérience quels sont les éléments sûrs à omettre pour votre configuration d'imagerie particulière.

Pour toutes les options de génération de PSF ci-dessus, l'argument optionnel **-ks=** peut être fourni pour définir la dimension de la PSF, et l'argument optionnel **-savepsf=nom de fichier** peut être utilisé pour sauvegarder la PSF générée : un nom de fichier doit être fourni et les mêmes exigences d'extension de nom de fichier s'appliquent que pour **makepsf save filename**

Liens : *psf*, *rl*, *sb*, *wiener*

Ligne de commande Siril

```
rl [-loadpsf=] [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Restaure une image en utilisant la méthode Richardson-Lucy.

En option, une PSF peut être chargé en utilisant l'argument **-loadpsf=filename** (créé avec MAKEPSF).

Le nombre d'itérations est donné par **-iters** (par défaut 10).

Le type de régularisation peut être défini avec **-tv** pour la variation totale, ou **-fh** pour la norme de Frobenius de la matrice hessienne (la valeur par défaut est aucune) et **-alpha=** fournit la force de régularisation (une valeur plus faible = plus de régularisation, par défaut = 3000).

Par défaut, la méthode du gradient descendant est utilisé avec une taille de pas par défaut de 0.0005, cependant la méthode multiplicative peut être spécifiée avec **-mul**.

Le critère d'arrêt peut être activé en spécifiant une limite d'arrêt avec **-stop=**

Liens : *psf*, *makepsf*

Ligne de commande Siril

```
sb [-loadpsf=] [-alpha=] [-iters=]
```

Restaure une image en utilisant la méthode Split Bregman.

En option, une PSF peut être chargée en utilisant l'argument **-loadpsf=nom de fichier**.

Le nombre d'itération est donné par **-iters** (par défaut 1).

Le facteur de régularisation **-alpha=** fournit la force de la régularisation (plus petite valeur = plus de régularisation, par défaut = 3000)

Liens : *psf*

Ligne de commande Siril

```
wiener [-loadpsf=] [-alpha=]
```

Restore une image en utilisant la méthode de déconvolution de Wiener.

En option, une PSF créée par MAKEPSF peut être chargée en utilisant l'argument **-loadpsf=filename**.

Le paramètre **-alpha=** fournit le facteur de régularisation modélisé par le bruit Gaussien

Liens : *psf*, *makepsf*

Références

10.4.6 Filtres de préservation des bords

Siril offers two edge preserving filters : OpenCV's implementation of the Bilateral Filter, and a Guided Filter. Both of these filters can be used to reduce noise, and both preserve sharp edges and strong gradients in the image. The bilateral filter acts on a single image whereas the guided filter filters the image using a guide image to identify structures. In the simplest use, the guided filter acts in a self-guided manner by using the input image as the guide image.

La présentation de la fenêtre de dialogue est assez simple et peu de paramètres sont disponibles.

- **Filter type** : Choose between the bilateral filter and the guided filter.
- **Diameter** : This defines the size of the filter kernel that will be used. When using the bilateral filter, setting a diameter of 0 will cause the filter kernel size to be set automatically based on the spatial sigma value. When using the guided filter this value must be set : a diameter of 0 will result in no filtering being done.
- **Sigma (spatial)** : This defines the spatial extent of the filter kernel. A larger value results in smoothing of noise over a greater image area, but is slower to compute. A smaller value results in more local smoothing and is faster to compute. Defaults are not set for the sigma values as the appropriate value may depend significantly on image characteristics, but it can be good to start with both sigma (spatial) and sigma (intensity) set to about 11.
- **Sigma (intensity)** : This defines the range of intensity variation to which the filter responds. A high value results in stronger filtering of noise but may result in loss of genuine detail ; a low value reduces the filtering of noise, but will avoid loss of details with gentler gradients.
- **Guide image** : This allows selection of an image to use as a guide image when performing a guided filter. If the "self guided" check box is checked, the filter acts in self-guided mode using the input image as the guide image.
- **Modulation** : In Siril, 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 edge-preserving filter at all.

Astuce

The parameters to the two different filter types do not behave identically, so when changing between bilateral and guided filters with sigma parameters set, you should expect to see some change in the preview result. The code applies a bit of compensation to the parameters provided to minimize the difference in behaviour of the two filters for the same input parameters but this is not exact (nor is it intended to be).

Ligne de commande Siril

```
epf [-guided] [-d=] [-si=] [-ss=] [-mod=] [-guideimage=]
```

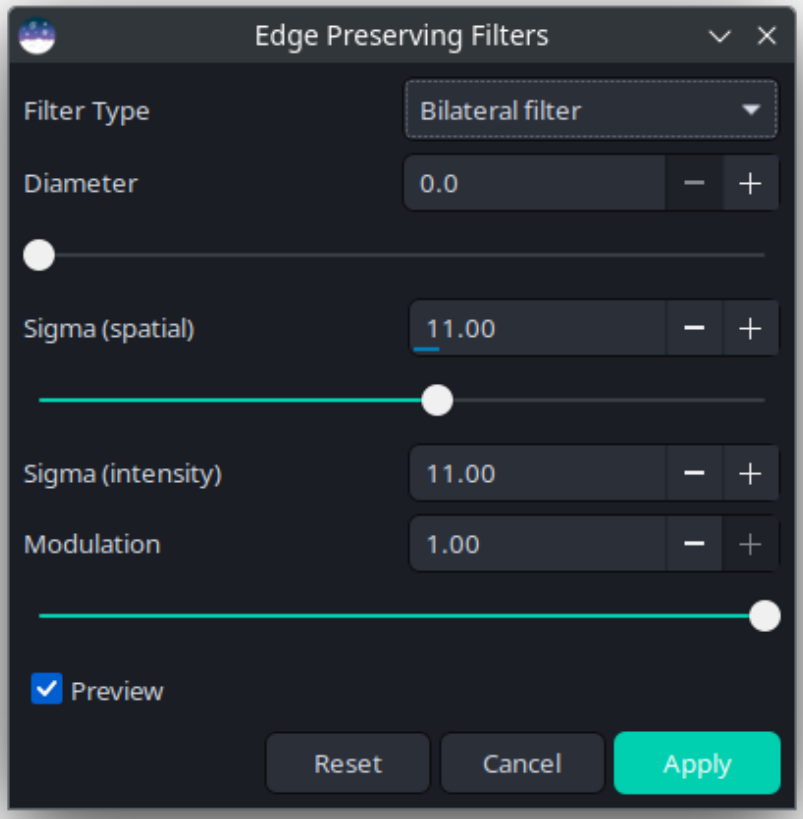


FIG. 48 – Fenêtre de dialogue Filtre Médian.

FIG. 49 – Showing the effect of changing the bilateral filter parameters. Note that the exact values required will depend on the noise characteristics of your data.

Applique un filtre préservant les bords. Par défaut, un filtre bilatéral est appliqué ; un filtre guidé peut être spécifié en utilisant l'argument **-guided**. Le diamètre du filtre est fixé par défaut à 3 et peut être défini à l'aide de **-d=**. Attention aux valeurs de d supérieures à 20, car l'algorithme peut être coûteux en termes de calcul.

La valeur sigma du filtrage d'intensité peut être définie à l'aide de **-si=** et la valeur sigma spatiale peut être définie à l'aide de **-ss=**. Les valeurs sigma représentent la différence entre les valeurs des pixels sur lesquelles le filtre agit fortement : pour les images 32 bits, la valeur doit être comprise entre 0 et 1,0, tandis que pour les images 16 bits, elle doit être comprise entre 0 et 65535. Par défaut, si rien n'est spécifié, les deux valeurs sont fixées à 11. Si **-d=0** est défini, le diamètre du filtre sera défini automatiquement en fonction de la valeur de **-ss**. *Notez que lors de l'application d'un filtre guidé, seul *-sc** s'applique.*

Lors de la spécification d'un filtre guidé, une image guide peut être définie en utilisant **-guideimage=**. Par défaut, si aucune image guide n'est spécifiée, un filtre auto-guidé est effectué. *Note : l'image guide doit avoir les mêmes dimensions que l'image à filtrer !*

L'intensité du filtre peut être modulée à l'aide de l'argument **-mod=**. Si mod = 1.0, l'effet complet du filtre sera appliqué ; pour mod inférieur à 1.0, une proportion de l'image originale sera mélangée au résultat, et pour mod = 0.0, aucun filtrage ne sera appliqué

10.4.7 Transformée de Fourier

Une transformée de Fourier (TF) est une transformation mathématique qui décompose les fonctions en composantes de fréquence, qui sont représentées par la sortie de la transformée en fonction de la fréquence. Cette transformation est largement utilisée en imagerie car elle permet de voir des signaux à des fréquences régulières.

Théorie

Transformée de Fourier

La transformée de Fourier est un processus d'analyse qui décompose une fonction à valeurs complexes $f(x)$ en ses fréquences constitutives et leurs amplitudes :

$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \xi x}\, dx.$

Transformation Inverse

Le processus inverse est la synthèse, qui recrée $f(x)$ à partir de sa transformation :

$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{2\pi i \xi x}\, d\xi.$

Siril permet de transformer une image dans l'espace des fréquences grâce à un algorithme de Transformation de Fourier rapide. Le résultat se présente sous la forme de deux images. La première, automatiquement chargée, contient le module de la transformation, la seconde contient la phase. L'emplacement des deux images doit être saisi dans l'onglet **Transformation directe** (voir illustration ci-dessous) de la boîte de dialogue. Il est ensuite possible de modifier l'image du module en supprimant les pics de fréquence correspondant aux signaux indésirables. Il est important de ne pas oublier de sauvegarder les modifications.

L'option *Centré*, lorsqu'elle est cochée, centre l'origine de la transformée de Fourier directe. Si elle n'est pas cochée, l'origine est située dans le coin supérieur gauche.

Pour reconstruire l'image, cliquez sur l'onglet **Transformation inverse** et entrez le chemin d'accès aux images de module et de phase.

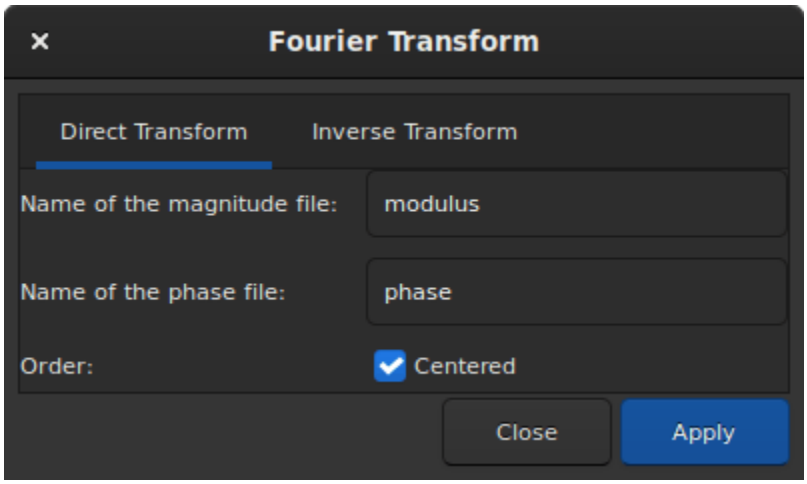


FIG. 50 – Onglet Transformation Direct.

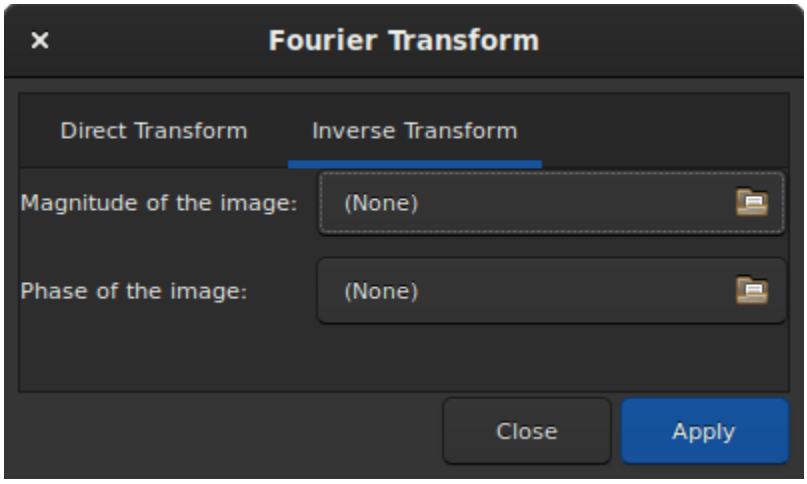


FIG. 51 – Onglet Transformation Inverse.

Ligne de commande Siril

```
fftd modulus phase
```

Applique une transformée de Fourier rapide à l'image chargée. **modulus** et **phase** donnés en argument sont les noms des fichiers FITS sauvegardés

Ligne de commande Siril

```
ffti modulus phase
```

Récupère l'image corrigée en appliquant une transformation inverse. Les arguments **modulus** et **phase** sont les noms des fichiers d'entrée, le résultat sera la nouvelle image chargée

10.4.8 Filtre Médian

La médiane représente le point central des données, la moitié des données étant plus petite et l'autre moitié plus grande que ce point. Il s'agit d'un estimateur robuste qui permet d'éliminer les valeurs aberrantes d'un ensemble de données. Par conséquent, cet outil peut être utile en tant que réducteur de bruit naïf, efficace contre le *bruit impulsionnel*.

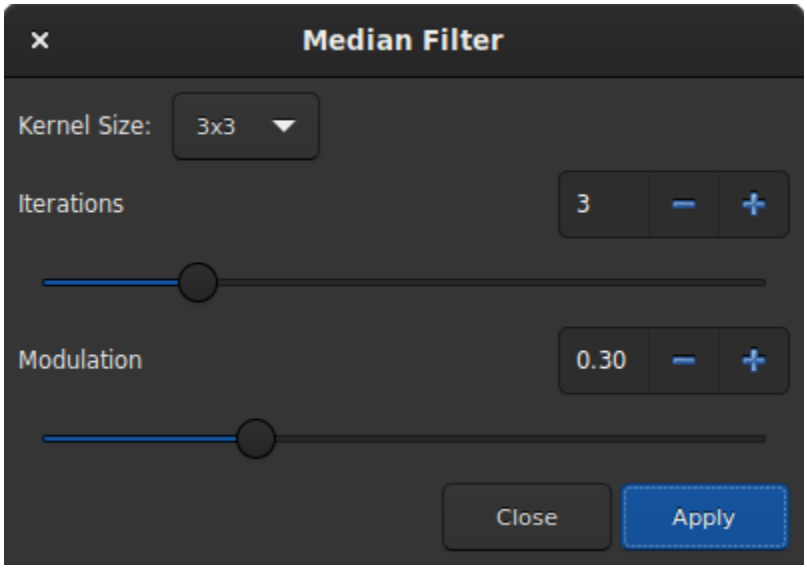


Fig. 52 – Fenêtre de dialogue Filtre Médian.

La présentation de la fenêtre de dialogue est assez simple et peu de paramètres sont disponibles.

— **Taille du noyau** : De 3×3 à 15×15, ceci définit la taille d'un noyau quadratique qui est utilisé pour appliquer le filtre. Plus le noyau est grand, plus le résultat sera flou.
— **Itérations** : Ce paramètre définit le nombre de passages du noyau.
— **Modulation** : Dans Siril, la modulation est un paramètre compris entre 0 et 1 qui mélange l'image originale et l'image traitée. Une valeur de 1 ne conserve que l'image traitée, une valeur de 0 n'applique aucun filtre médian.

Ligne de commande Siril

```
fmedian ksize modulation
```

Effectue un filtre médian de taille **ksize** x **ksize** (**ksize** DOIT être impair) sur l'image chargée avec un paramètre de modulation **modulation**.

Le pixel de sortie est calculé comme suit : out = mod x m + (1 mod) x in, où m est la valeur du pixel filtré par médiane. Une valeur de modulation de 1 n'appliquera aucune modulation

10.4.9 Réduction de bruit

Bruit de l'image

Les images souffrent de différents types de bruit :

1. Le bruit impulsionnel
   — Ce type de bruit (parfois appelé “bruit poivre et sel”) provient généralement de pixels chauds ou froids. Il est généralement traité en utilisant l’empilement par rejet, mais vous devrez parfois y faire face si vous traitez une image seule, non empilée.
2. Bruit blanc gaussien additif
   — Ce type de bruit est typique des photographies bien éclairées : il provient des fluctuations thermiques et électroniques du dispositif d’acquisition, et le niveau de bruit est indépendant du signal. Il peut être réduit au moment de la capture en utilisant des caméras refroidies, et il est réduit lors de l’empilement car l’empilement de n images augmente le signal corrélé d’un facteur n alors que le bruit non corrélé n’augmente que d’un facteur $\sqrt{n}$. C’est également le type de bruit que la plupart des algorithmes de débruitage classiques sont conçus pour éliminer.
3. Bruit de Poisson
   — Lorsque l’on a affaire à des images pauvres en photons, le caractère du bruit cesse d’être principalement gaussien et la nature probabiliste du comptage des photons devient significative, voire dominante. Ceci est modélisé par une distribution de Poisson et ce type de bruit dépend du signal.

Réduction du bruit dans Siril

Siril fournit des algorithmes classiques de débruitage bien étudiés et à la pointe de la technologie. Les critères qui nous ont poussé à choisir les algorithmes présentés ici étaient les suivants :

— L’algorithme devait être analysé dans des revues académiques à comité de lecture, avec une description de l’algorithme et une comparaison quantitative objective de ses performances.
— Les auteurs doivent avoir mis à disposition une implémentation F/OSS. Ceci est important pour éviter les problèmes de propriété intellectuelle et, lorsque les implémentations de référence ont été utilisées directement, pour assurer la compatibilité des licences.
— Les algorithmes doivent fonctionner à une vitesse raisonnable.
— Enfin, l’implémentation de l’algorithme doit être capable de traiter des données de pixels en virgule flottante 32 bits.

La technologie de débruitage par réseau neuronal a été étudiée, mais écartée pour le moment en raison de la complexité de son développement. La performance de débruitage des réseaux neuronaux peut typiquement battre les approches classiques jusqu’à un dB de rapport signal/bruit de pointe, mais la performance dépend fortement du réseau neuronal entraîné sur des données représentatives des données réelles.

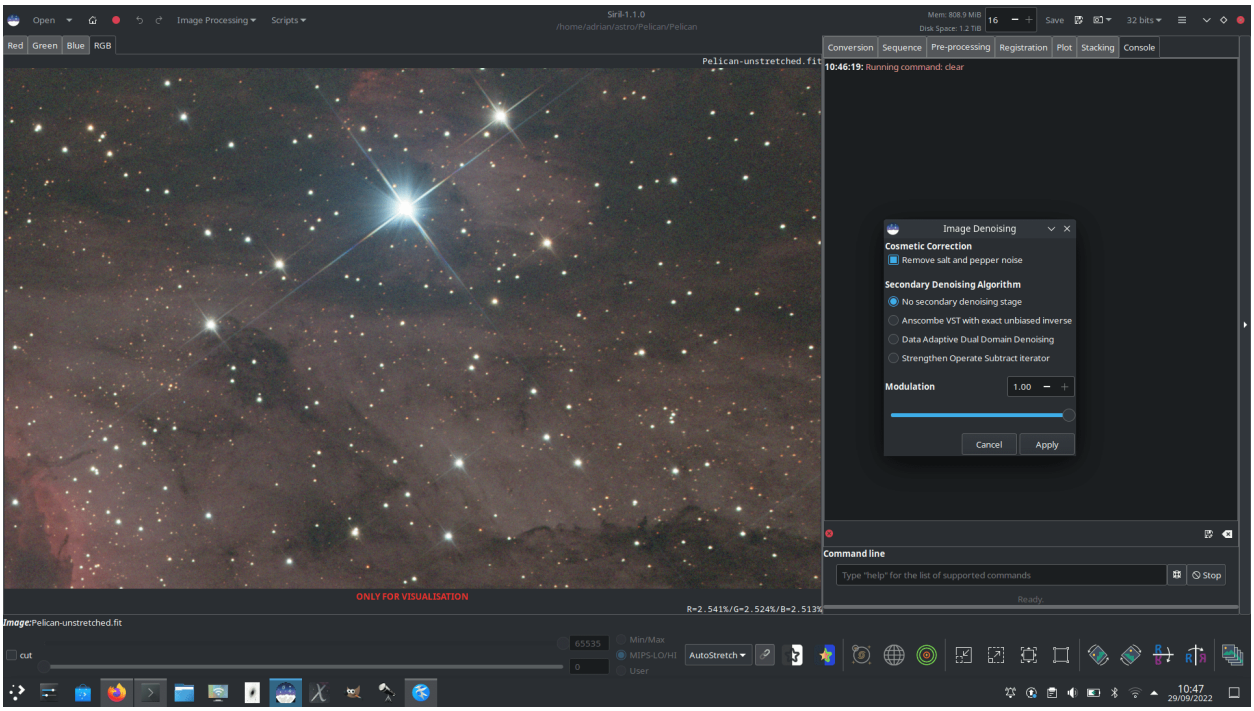


Fig. 53 – Fenêtre du réduction de bruit

Algorithmes : Bruit impulsionnel

Siril élimine principalement le bruit d'impulsion par l'empilement avec rejet sigma. Si vous utilisez cette méthode d'empilement, vous ne devriez pas avoir de problèmes avec le bruit impulsionnel. Cependant, si vous travaillez sur une seule exposition, vous pouvez trouver du bruit d'impulsion dans votre image. Celui-ci doit être traité à l'aide de la fonction **Correction cosmétique** de Siril avant toute autre réduction de bruit, car la présence de bruit impulsionnel peut fausser les algorithmes de réduction de bruit et créer des artefacts. Elle fonctionne de la même manière que le rejet sigma, mais sur les pixels voisins. Tout pixel dont l'intensité est éloignée de plus de *n* écarts types de ses voisins sera rejeté et remplacé par une valeur basée sur la médiane des voisins. Dans l'outil de réduction de bruit, la correction cosmétique est active par défaut et intervient avant toute autre étape de réduction de bruit. (Même si la suppression du bruit d'impulsion a déjà été effectuée, il n'y a pas d'inconvénient à laisser ce paramètre activé). La correction cosmétique peut également être appliquée manuellement en utilisant l'outil *Correction cosmétique* dans le menu *Traitement de l'image*.

Algorithmes : Bruit blanc gaussien additif

Le principal algorithme de réduction du bruit blanc Gaussien additif utilisé dans Siril est le débruitage bayésien non local (NL-Bayes) [Lebrun2013].

— Les algorithmes de débruitage non local ont représenté une amélioration majeure par rapport aux précédents filtres linéaires centrés sur le pixel. NL-Bayes est une version améliorée des algorithmes de débruitage non local précédents et offre l'un des meilleurs algorithmes classiques de débruitage de bruit blanc Gaussien. Il est légèrement meilleur que l'algorithme moderne "de référence" de réduction du bruit Block Matching and 3D tranform (BM3D) et beaucoup plus rapide à exécuter.

— Le paramètre clé requis pour optimiser les performances des algorithmes de débruitage du bruit blanc gaussien additif est le paramètre sigma : l'écart-type du bruit. Siril mesure le niveau de bruit directement à partir des données de l'image et le transmet à l'algorithme NL-Bayes. Par conséquent, dans l'outil de débruitage de Siril, il n'y a rien à configurer pour NL-Bayes, et ce dernier s'utilise très simplement.

Siril fournit en complément à NL-Bayes un certain nombre d'autres algorithmes de réduction du bruit :

— Débruitage adaptatif à double domaine (DA3D) [Pierazzo2017]

  - Cet algorithme prend la sortie de NL-Bayes et l'utilise comme image guide. Cette image guide est utilisée pour retraiter l'image originale en effectuant un rétrécissement dans le domaine fréquentiel sur des patchs adaptés à la forme et aux données. Cela améliore légèrement les performances de NL-Bayes avec un coût de calcul supplémentaire. Les patchs adaptés à la forme et aux données sont sélectionnés dynamiquement, ce qui concentre les calculs sur les zones les plus détaillées de l'image. Cette méthode peut également contribuer à réduire les artefacts en forme d'escalier présents dans l'image guide.
  - Dans l'outil de débruitage de Siril, DA3D est un simple bouton sans paramètres optionnels.
- L'itération Renforcer, Opérer, Soustraire (appelé ici itération SOS) [Romano2015]
  - SOS fonctionne en itérant l'algorithme de débruitage primaire plusieurs fois. À chaque itération, l'image est "renforcée" par l'ajout d'une proportion de l'image bruitée d'origine. L'algorithme NL-Bayes est exécuté sur cette image renforcée, après quoi l'estimation précédente est soustraite.
  - L'image x à une itération $k+1$ est donnée par $x_{k+1} = f(y + x_k) - x_k$ où y est l'image bruitée d'entrée.
  - Dans l'outil de débruitage de Siril, SOS est composé d'un bouton avec deux paramètres : le nombre d'`itérations` peut être réglé, et la proportion de l'image bruitée mélangée à chaque itération (`rho`) peut être définie. Évitez de régler `rho` trop haut car cela peut entraîner des problèmes de convergence de l'algorithme SOS : les valeurs par défaut (`3 itérations` et `rho = 0.2`) sont généralement correctes.

Algorithmes : Bruit de Poisson et Poisson-Gaussien

- Transformée stabilisatrice de variance d'Anscombe [Mäkitalo2011], [Mäkitalo2012]
  - Les transformées de stabilisation de la variance sont utilisées pour les images présentant un bruit de Poisson ou de Poisson-Gaussien afin de minimiser la dépendance du bruit vis-à-vis du signal et de le faire ressembler davantage à un bruit blanc gaussien, que NL-Bayes réussit bien à éliminer, puis une transformée inverse est appliquée à la fin. La transformation choisie pour Siril est la transformation d'Anscombe$A : x \rightarrow 2 \times \sqrt{(x + \frac{3}{8})}$
  - Comme la transformation n'est pas linéaire, l'utilisation de l'inverse algébrique direct entraîne un biais en sortie. Siril utilise donc une approximation à forme fermée de l'inverse exact non biaisé, qui est rapide à calculer et produit une amélioration substantielle par rapport à d'autres formes d'inverse telles que l'inverse asymptotique.
  - Dans l'outil de débruitage de Siril, la VST d'Anscombe est un simple bouton sans aucun paramètre optionnel.

Notez que seul un des algorithmes de débruitage complémentaires mentionnés ci-dessus peut être choisi à la fois.

L'animation ci-dessous montre ce qu'il est possible de faire en utilisant la stabilisation de la variance avec une image pauvre en photons, dans ce cas une seule image de la nébuleuse du Pélican de 5min prise avec un filtre rouge, montrée avec la fonction de transfert d'écran *Auto Ajustement*. Notez l'absence de flou, de gonflement ou de perte de détails autour des étoiles et le bord net de la nébuleuse dans la partie inférieure gauche de l'image par rapport à ce que l'on pourrait obtenir avec des schémas de réduction du bruit plus basiques. Une fois étiré de façon plus sympathique et combiné avec d'autres canaux, cela améliorerait grandement la qualité réalisable à partir de données très limitées (bien que l'accumulation de données soit toujours la meilleure solution pour réduire le bruit d'une image empilée !)

FIG. 54 – Débruitage d'une image pauvre en photon

Modulation

La modulation consiste à mélanger une proportion de l'image bruitée d'origine dans le résultat final. Cela réduit évidemment les performances du débruitage, mais dans certains cas, si le débruitage a laissé des zones plates de l'image un peu trop lisses et non naturelles, vous pouvez utiliser la modulation pour restaurer l'apparence de la microtexture dans ces régions.

Quand utiliser la réduction du bruit

Les algorithmes de réduction du bruit sont conçus pour éliminer le bruit blanc gaussien et devraient donc être plus performants sur des images non étirées : si un bruit blanc est étiré de manière non linéaire, ses caractéristiques changent et il n'est plus blanc. Il est toujours possible d'effectuer une réduction du bruit sur des images étirées et d'obtenir une amélioration, mais elle ne sera peut-être pas aussi efficace que si elle était appliquée au stade linéaire.

Interface de réduction du bruit

L'outil de réduction du bruit Siril est accessible de deux manières : via l'interface graphique ou via la ligne de commande. L'interface graphique est présentée ci-dessous. Remarque : les options avancées de SOS sont masquées si SOS n'est pas sélectionné.

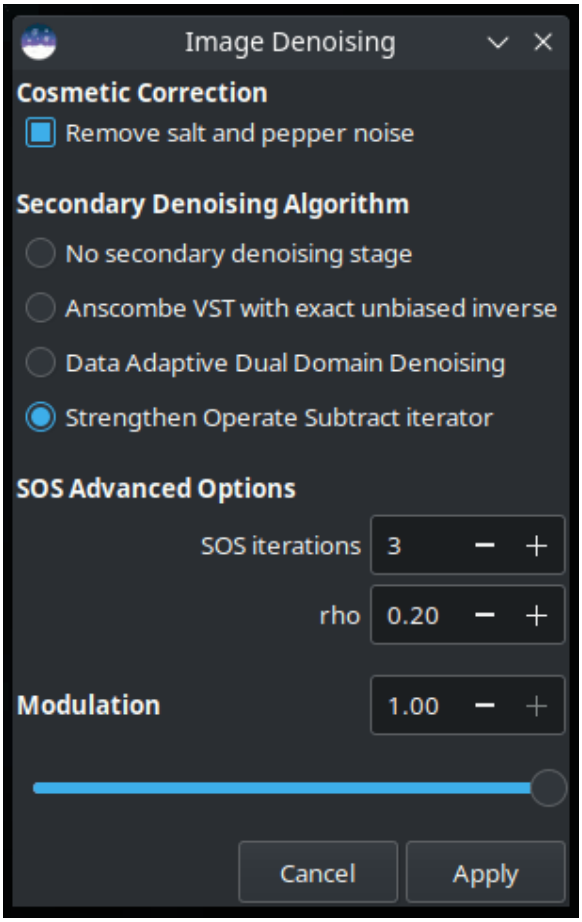


Fig. 55 – Interface graphique de la réduction de bruit de Siril

La réduction du bruit peut également être appliquée à l'aide de la ligne de commandes de Siril, soit dans la console, soit dans des scripts. Le format est le suivant :

Ligne de commande Siril

```
denoise [-nocosmetic] [-mod=m] [ -vst | -da3d | -sos=n [-rho=r] ] [-indep]
```

Réduit le bruit de l'image en utilisant l'algorithme bayésien non local décrit par Lebrun, Buades et Morel.

Il est fortement recommandé d'appliquer une correction cosmétique pour éliminer le bruit sel et poivre avant d'exécuter le réducteur de bruit, et par défaut cette commande applique automatiquement la correction cosmétique. Toutefois, si celle-ci a déjà été effectuée plus tôt dans le flux de travail, elle peut être désactivée ici à l'aide de la commande optionnelle **-nocosmetic**.

Un argument optionnel **-mod=m** peut être donné, où 0 <= m <= 1. Le pixel de sortie est calculé comme suit : *out=m x d + (1 m) x in*, où *d* est la valeur de pixel réduite du bruit. Une valeur de modulation de 1 n'appliquera aucune modulation. Si le paramètre est omis, il est par défaut à 1.

L'argument optionnel **-vst** peut être utilisé pour appliquer la transformation de stabilisation de variance Anscombe généralisée avant NL-Bayes. Cela est utile avec des images à faible luminosité telles que des images simples, où le bruit suit une distribution de Poisson ou Poisson-Gaussien plutôt qu'une distribution principalement gaussienne. Il ne peut pas être utilisé en conjonction avec DA3D ou SOS, et pour le débruitage d'images empilées, il n'est généralement pas bénéfique.

L'argument optionnel **-da3d** peut être utilisé pour activer le débruitage Data-Adaptive Dual Domain (DA3D) en tant qu'algorithme de débruitage final. Cela utilise la sortie de BM3D comme image de guidage pour affiner le débruitage. Il améliore les détails et réduit les artefacts en escalier.

L'argument optionnel **-sos=n** peut être utilisé pour activer le renforcement itératif de la détection de bruit Strengthen-Operate-Subtract (SOS), avec le nombre d'itérations spécifié par n. En particulier, ce renforceur peut produire de meilleurs résultats si l'algorithme NL-Bayes non renforcé produit des artefacts dans les zones de fond de ciel. Si à la fois -da3d et -sos=n sont spécifiés, le dernier spécifié sera appliqué.

L'argument facultatif **-rho=r** peut être spécifié, où 0 < r < 1. Cela est utilisé par le booster SOS pour déterminer la quantité d'image bruitée ajoutée au résultat intermédiaire entre chaque itération. Si -sos=n n'est pas spécifié, le paramètre est ignoré.

Par défaut, DA3D ou SOS ne sont pas appliqués, car l'amélioration de la réduction de bruit est généralement relativement faible et ces techniques nécessitent un temps de traitement supplémentaire.

En de très rares occasions ,lors du débruitage des images en couleur, des artefacts de couleur en bloc peuvent apparaître. L'argument optionnel **-indep** peut être utilisé pour éviter cela en effectuant le débruitage de chaque canal séparément. Cela est plus lent mais éliminera les artefacts

Comparaison

Les images ci-dessous fournissent une comparaison simpliste des différents algorithmes. Notez qu'une seule image est utilisée : en pratique, les différents algorithmes seront mieux adaptés à des images différentes. Vous pouvez cliquer sur toutes les images pour les visualiser avec un zoom de 100 %.

Image originale bruitée

Débruitage avec NL-Bayes uniquement

Débruitage avec NL-Bayes uniquement, avec une modulation de 75% pour restaurer une certaine microtexture

Débruitage avec NL-Bayes en utilisant la transformée d'Anscombe

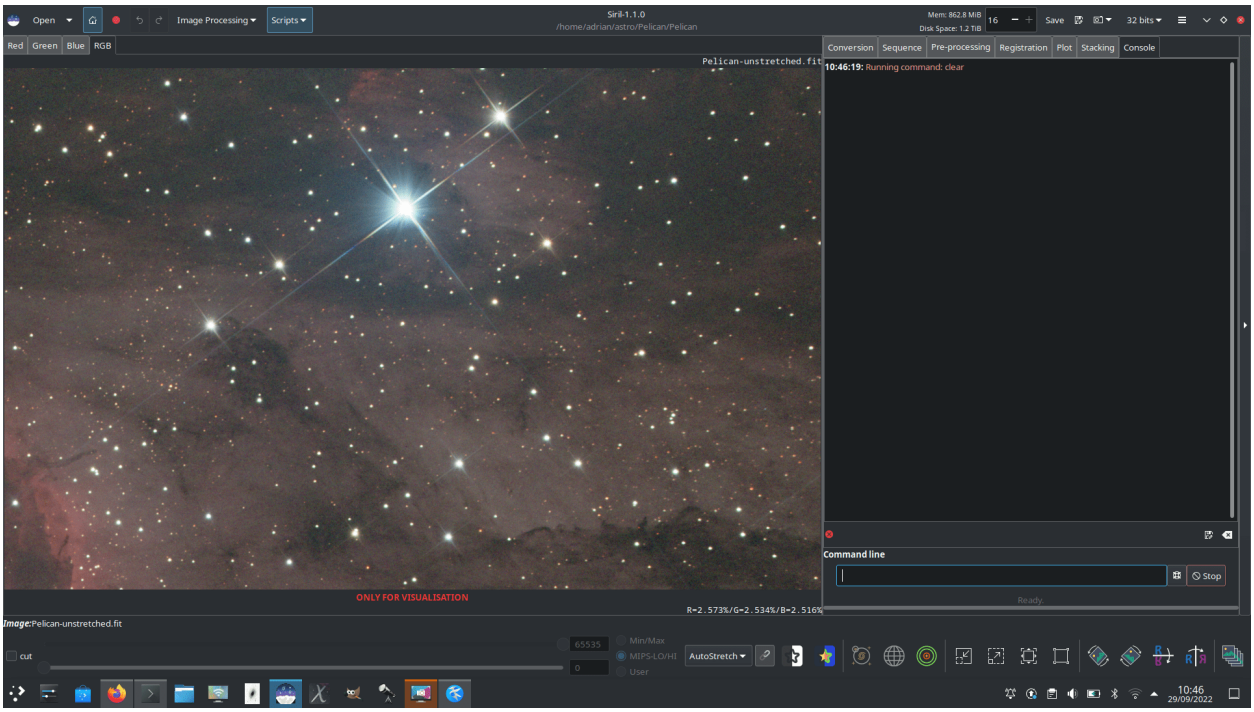


FIG. 56 – Imagebruitée

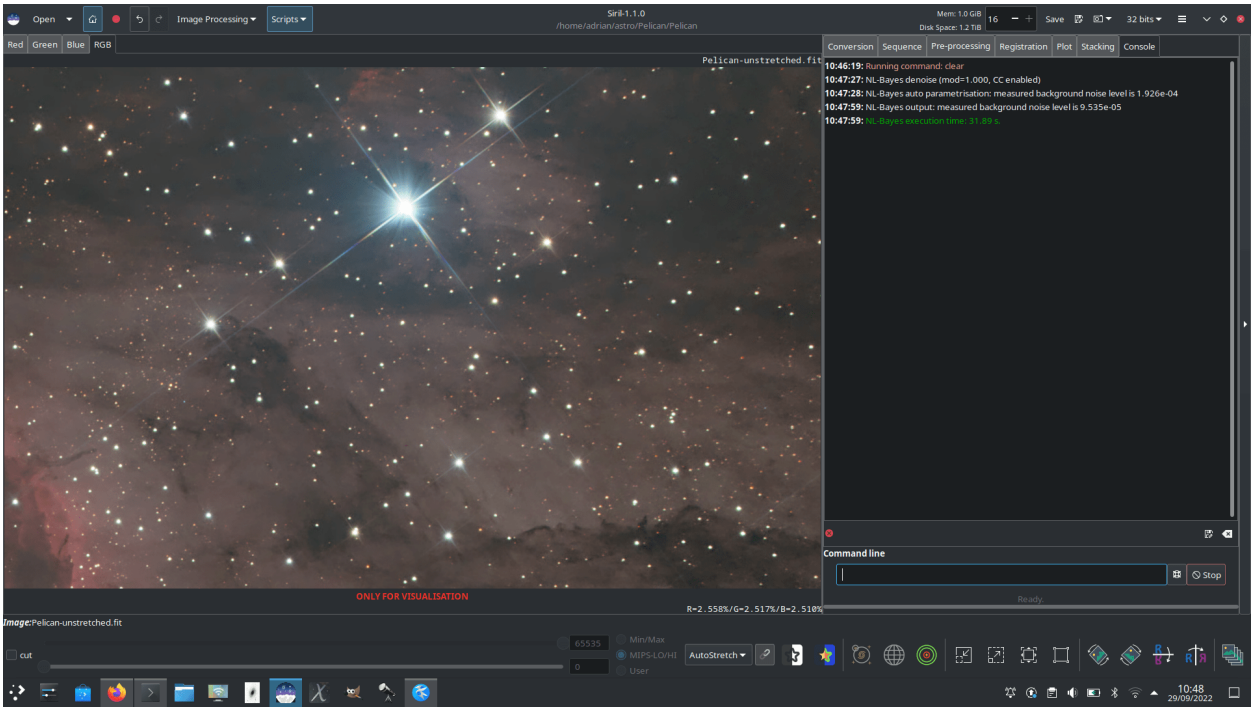


FIG. 57 – Débruitage avec NL-Bayes uniquement

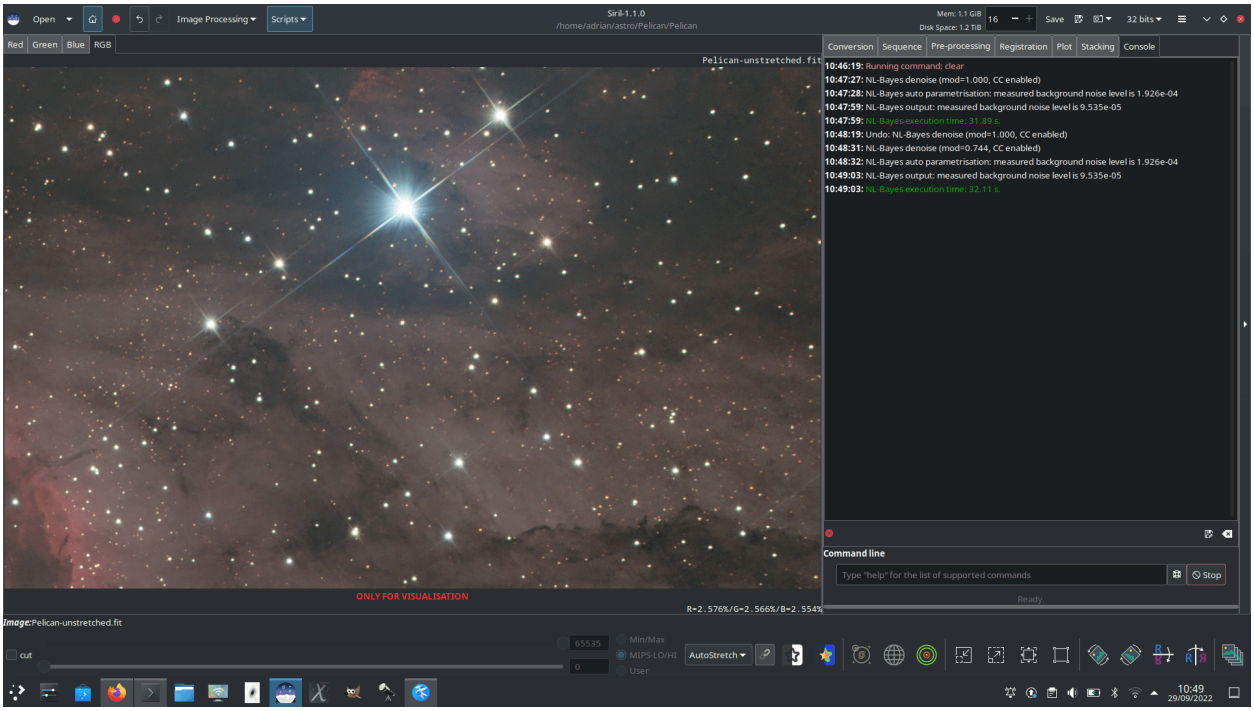


Fig. 58 – Utilisation de la modélisation

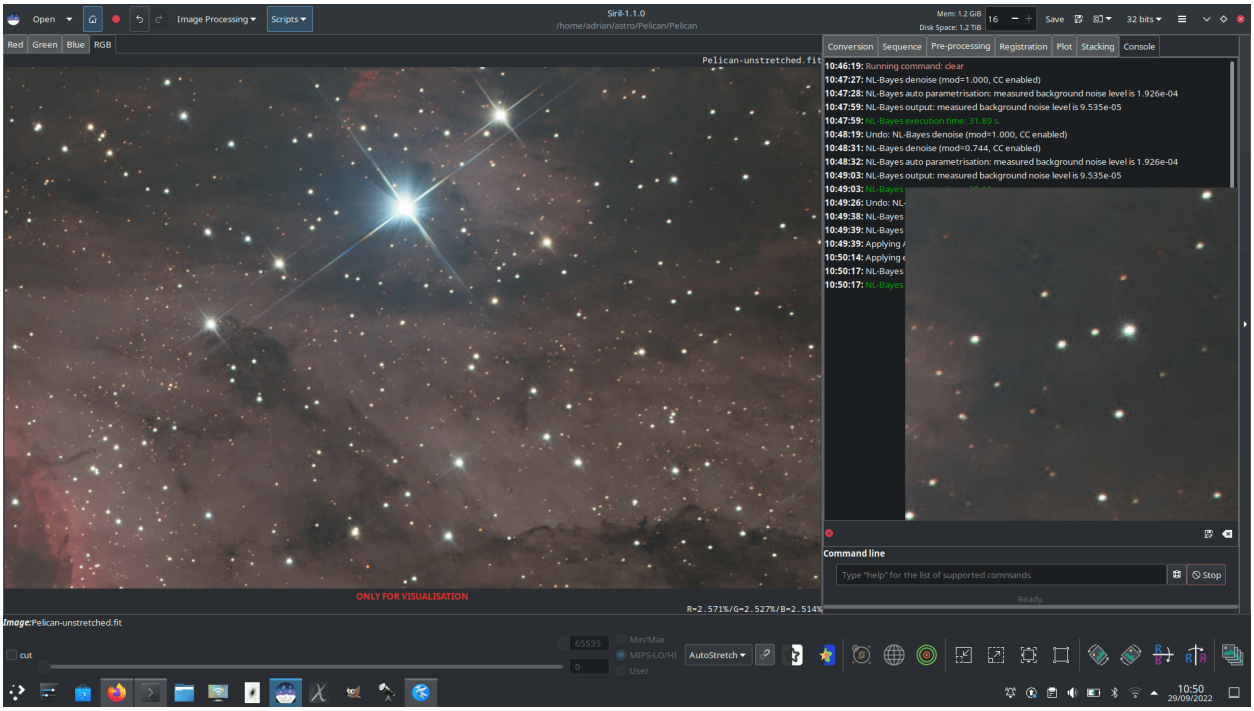


Fig. 59 – Débruitage avec NL-Bayes, stabilisation de la variance avec la transformée d'Anscombe. Un zoom non interpolé de 200% est montré à droite.

Débruitage avec DA3D en utilisant une image guide NL-Bayes

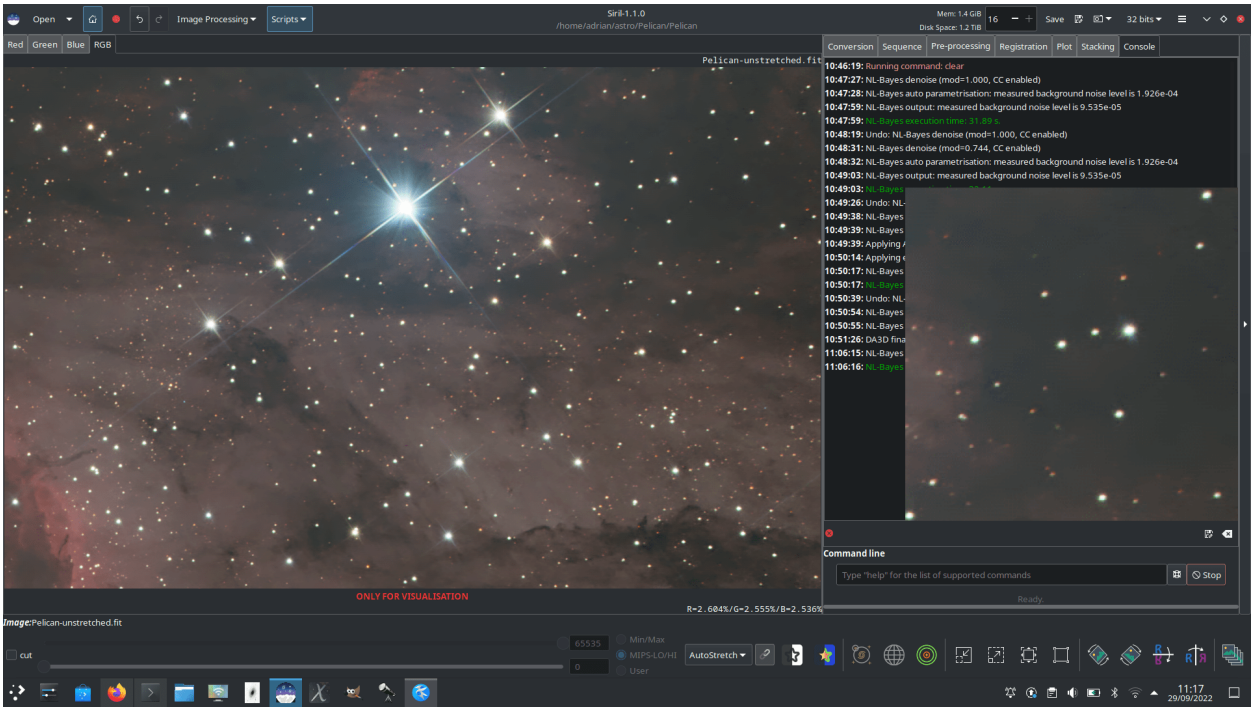


FIG. 60 – Débruitage avec DA3D, image guide préparée avec NL-Bayes. Un zoom non interpolé de 200% est montré à droite.

Débruitage avec NL-Bayes et SOS

Limitations

La principale limitation est que les algorithmes fonctionnent mieux lorsque le bruit est de nature gaussienne (ou peut être rendu approximativement gaussien à l'aide du VST). Il y a quelques raisons pour lesquelles cela pourrait ne pas être vrai :

— Si l'image a déjà été fortement traitée, par exemple avec une déconvolution ou une accentuation par ondelettes, le caractère du bruit ne sera généralement plus gaussien. Si la réduction du bruit et la déconvolution font toutes deux partie de votre flux de travail, la réduction du bruit doit être effectuée en premier.

— Les images couleur peuvent être moins bien débruitées que les images couleur mono ou composites. Une petite réduction du bruit blanc gaussien de luminance est obtenue mais, suite au processus de débruitage, le caractère du bruit est modifié de sorte qu'il n'est plus bien modélisé comme bruit blanc gaussien et n'est pas éliminé très efficacement. De plus, pour les images monochromes OSC et composées, le bruit de chrominance tend à ne pas être bien modélisé comme bruit blanc gaussien et nécessite un traitement différent. À l'heure actuelle, le bruit de chrominance est mieux traité dans les logiciels de manipulation d'images à usage général tels que GIMP.

Références

10.4.10 Gradient rotationnel (filtre de Larson Sekanina)

Le gradient de rotation, également appelé filtre de Larson Sekanina, est un filtre qui permet de supprimer les structures circulaires d'une image, afin de mieux faire ressortir d'autres détails. Cette technique est particulièrement efficace pour montrer les jets sortant du noyau d'une comète.

Le principe est assez simple : ce traitement d'image consiste à soustraire deux copies de l'image l'une de l'autre, l'une des deux copies ayant été préalablement tournée par rapport à un point défini dans l'image.

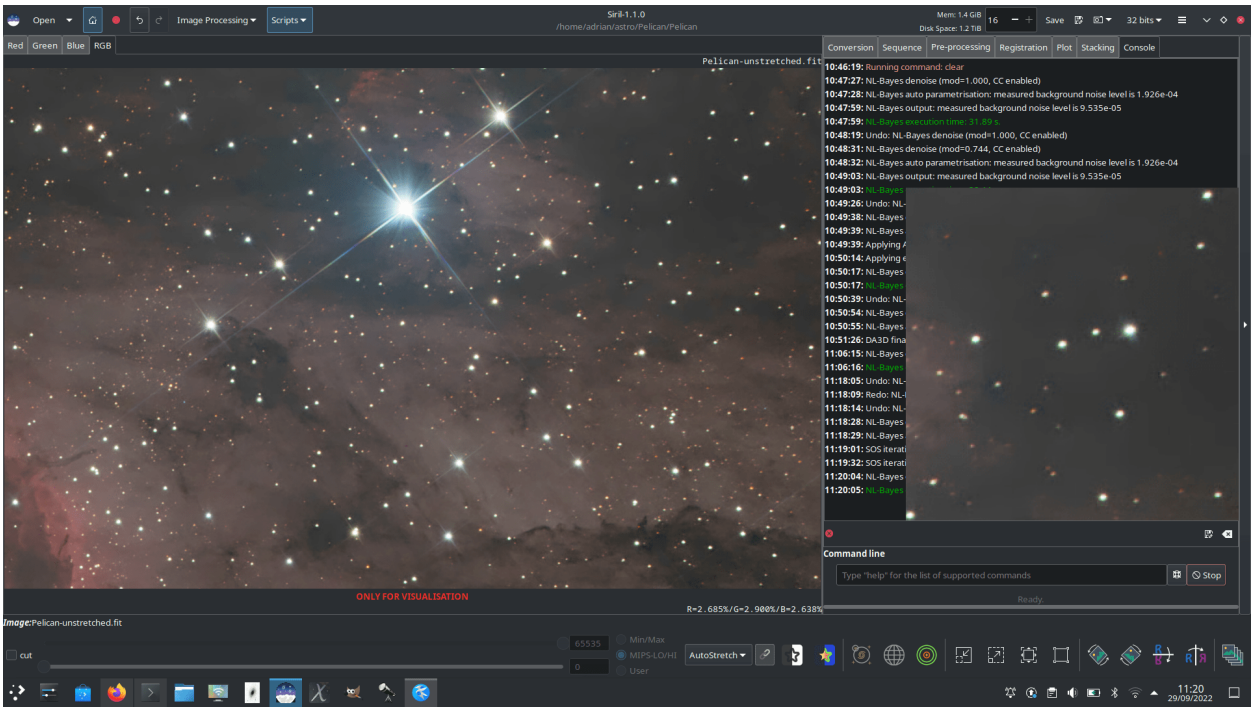


FIG. 61 – Débruitage avec des itérations NL-Bayes et SOS. Un zoom non interpolé de 200% est montré à droite.

— S'il existe des structures circulaires autour de ce point, elles ne sont pas modifiées par la rotation et disparaîtront après celle-ci.
— S'il existe des structures non circulaires, comme des jets dans la coma, elles seront décalées l'une par rapport à l'autre entre les deux copies et la soustraction amplifiera le contraste de cette structure dans le résultat.
— Si la comète se déplace dans l'image, il est possible d'ajouter un décalage radial.

Théorie

À partir d'une image d'entrée, le filtre génère deux images, chacune avec un décalage radial (dr en pixels) et un décalage en rotation ($d\alpha$ en degrés) par rapport au point (x_c,y_c). Ces décalages ont la même ampleur, mais des signes opposés entre les deux images. Les deux images sont ensuite combinées pour produire l'image finale. En coordonnées polaires(r, a) par rapport au point(x, y) nous avons :

$$B'(\alpha, r, d\alpha, dr) = 2 \cdot B(\alpha, r) - B(\alpha - d\alpha, r - dr) - B(\alpha + d\alpha, r - dr)$$

with B : the starting image, B' : the resulting image, dr : the radial shift and $d\alpha$: the rotational shift

Dans l'exemple ci-dessous, concernant la comète C/2022 E3, l'alignement a été fait sur la comète et les étoiles montrent des traînées importantes. La coma est très circulaire et il est difficile de voir les détails de son activité. Il n'est donc pas nécessaire de définir un décalage radial. Pour la rotation, un angle de 15° a été choisi (ce choix a été fait après plusieurs tentatives et en utilisant le bouton undo pour revenir en arrière). Pour choisir les coordonnées du centre de rotation, il suffit de faire une sélection autour du noyau cométaire et de cliquer sur *Utiliser la sélection courante*. Cette action copiera les coordonnées du centroïde à l'endroit désiré.

Un simple clic sur*Apply* permet d'appliquer le filtre. Dans notre exemple, un jet devient visible.

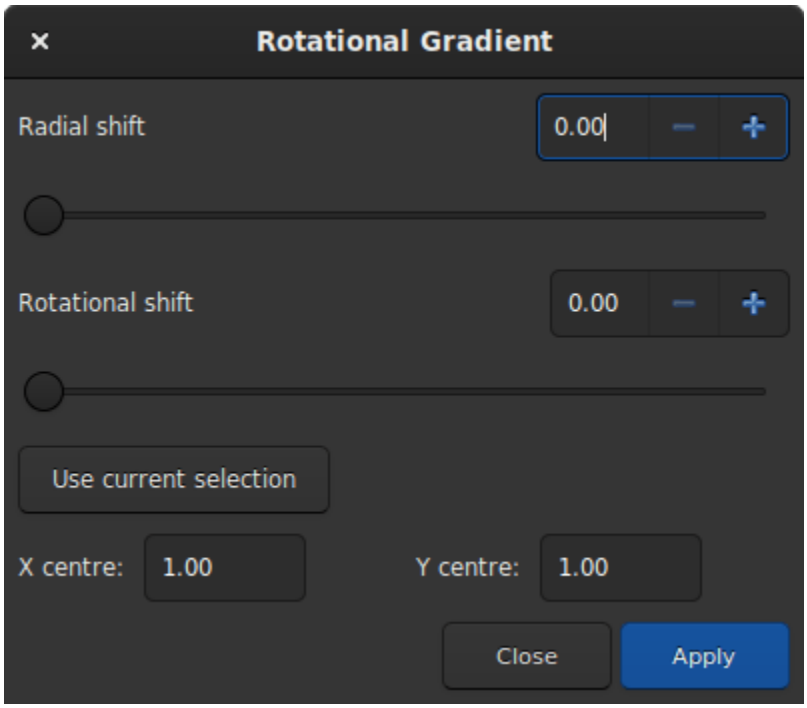


FIG. 62 – Boîte de dialogue du filtre Gradient de rotation.

FIG. 63 – Image d'une comète (C/2022 E3 (ZTF)) montrant une coma et une queue (avec l'aimable autorisation de Stéphane Garro).

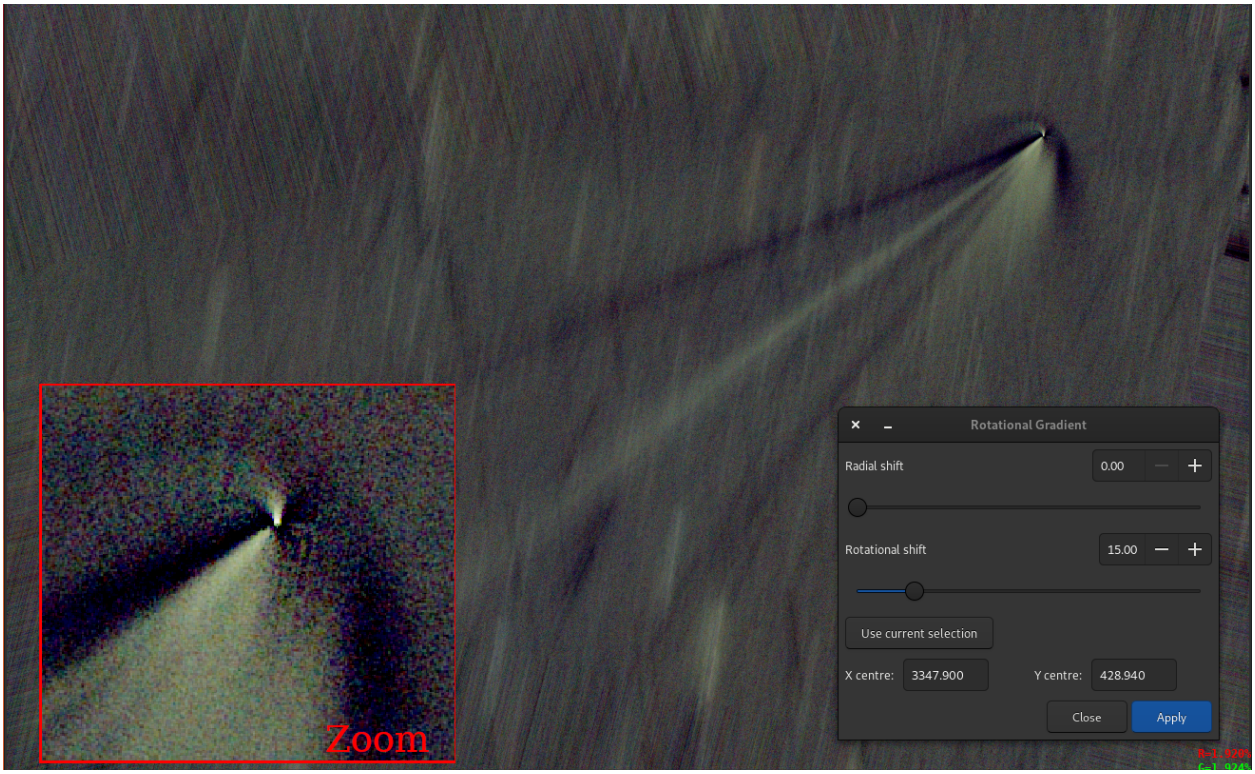


FIG. 64 – Après application du filtre, nous constatons qu'un jet émerge du noyau.

Ligne de commande Siril

```
rgradient xc yc dR dalpha
```

Crée deux images, avec un décalage radial (**dR** en pixels) et un décalage rotatif (**dalpha** en degrés) par rapport au point (**xc**, **yc**).

Entre ces deux images, le décalage a la même amplitude mais est de signe opposé. Ces deux images sont additionnées pour créer l'image finale. Ce processus est aussi appelé filtre Larson Sekanina

10.4.11 Filtre anti-violet

This is a cosmetic filter designed to reduce the effect of purple fringing around stars that may be visible when using an achromat refractor especially when broadband imaging. It operates on stretched colour images.

Astuce

This filter is designed to work on non-linear images. It is therefore imperative to use it once the histogram has been stretched.

Le filtre propose deux modes de fonctionnement : ses effets peuvent être globaux ou limités aux étoiles uniquement.

It reduces purple fringing by applying a luminance mask to the blue channel based on the green channel, and by allowing

you to reduce the blue channel intensity. When in starmask mode, it limits the effects to areas around identified stars.

Global

This mode is useful when working with a starmask or with images containing just stars, e.g. a globular cluster. This is not so useful for nebula or galaxy images where the colour might be affected.

To use, you should reduce the *Blue Adjustment* slider until the effects are optimal. This is generally around 0.15.

You may also increase the *Threshold* slider to restrict the effects to those pixels where the luminance is above this value. In this mode, the optimal value will be quite low.

Masque d'étoile

Ce mode est utile pour les images composites lorsque vous souhaitez limiter les effets de ce filtre aux étoiles uniquement.

To use, you should tick *Use Starmask*. If you've not already defined a starmask using the Dynamic PSF function, there will be a delay of a few seconds whilst the starmask is generated. You may now temporarily reduce the *Blue Adjustment* to zero. Your stars should now be yellow. If you still see purple fringing, you should increase the Threshold slider until the fringing is entirely covered by yellow. Now you can increase the *Blue Adjustment* slider until your stars are not yellow but your purple fringing is reduced or eliminated, typically around 0.12-0.15.

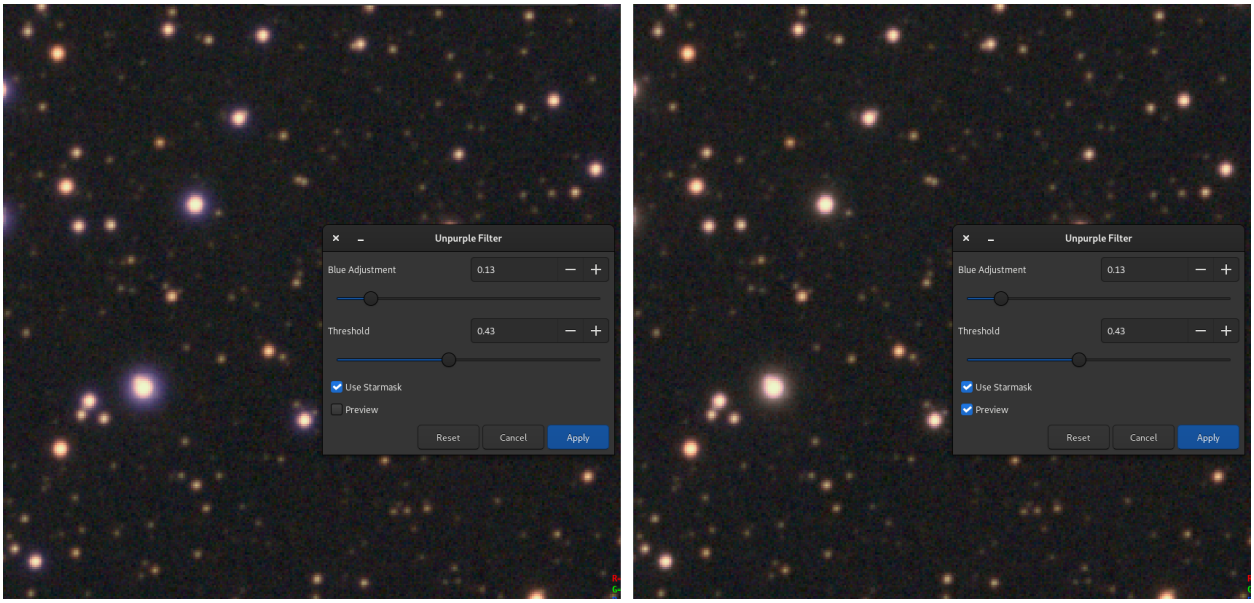


Fig. 65 – Comparison before/after application of the unpurple filter. We can see very clearly that the stars have a better color, without the purple ring.

Ligne de commande Siril

```
unpurple [-starmask] [-blue=value] [-thresh=value]
```

Applique un filtre cosmétique pour réduire les effets de franges violettes sur les étoiles.

If the **-starmask** parameter is given, a star mask will be used to identify areas of the image to affect. If a Dynamic PSF has already been run, this will be used for the starmask, otherwise one will be created automatically. The

-mod= parameter should be given a value somewhere around 0.14 to reduce the amount of purple. The **-thresh=** will specify the size modifier for each star in the starmask and should be large enough to cause the stars to be entirely processed without remaining purple fringing. The value should between 0 and 1, typically around 0.5.

If the **-starmask** parameter is not given, the purple reduction will be applied across the entire image for any purple pixels with a luminance value higher than the given **-thresh=**. In this case, the **-thresh=** value should be reasonably low. This mode is useful for starmasks or other images without nebula or galaxy

Liens : *psf*

10.5 Traitement des étoiles

Les étoiles font partie intégrante des images du ciel profond et jouent un rôle crucial pour faire ressortir la beauté et les détails des objets célestes. Elles apparaissent souvent comme des points lumineux brillants, mettant en valeur leur éclat et leurs couleurs, ce qui rend les images du ciel profond vraiment captivantes. Cependant, en raison des limites des conditions d'observation, les étoiles dans ces images peuvent apparaître plus grandes et surexposées. Pour y remédier, les astronomes utilisent des techniques avancées de traitement d'image pour traiter séparément les étoiles et contrôler leur taille et leur luminosité dans l'image finale.

Cette partie de la documentation est ensuite consacrée à tout ce qui concerne le traitement des étoiles.

10.5.1 Suppression des étoiles avec StarNet

Avertissement

The Starnet interface built into Siril itself is now deprecated. StarNet 2.5 has been released and is supported by means of a Python script which is available in the scripts repository. The legacy C interface will **not** support Starnet 2.5 but will remain available in Siril 1.4.x for users of older versions of Starnet.

In Siril 1.6 the legacy C interface will be removed so all users will need to upgrade to Starnet 2.5 and use the python script. The script does provide CLI args so where scripts use the Siril `starnet` command they will need to adapt to using `pyscript StarNet.py` instead.

StarNet est un logiciel développé par Nikita Misiura. Sa première version a été publiée sous une licence libre et gratuite. Malheureusement, la version 2 est devenue propriétaire et les sources ont été fermées depuis. La version 2 est disponible gratuitement à partir de là. Assurez-vous de télécharger la version **Ligne de Commande**. Siril peut s'interfacer avec n'importe quelle version de l'outil CLI de StarNet, y compris la nouvelle version expérimentale basée sur Torch qui a été initialement publiée pour les Macs Apple basés sur M1 et M2.

Avertissement

Si vous vous demandez **pourquoi StarNet ne se lance pas**, lancez d'abord le logiciel en dehors de Siril. Ce n'est pas la faute de Siril s'il n'est pas pris en charge par votre ordinateur ou s'il est mal installé pour une raison quelconque. Si votre processeur ne prend pas en charge les instructions de vectorisation requises par StarNet, il n'y a aucun moyen de contourner ce problème. Le message d'erreur sera obtenu en exécutant StarNet seul.

Astuce

Sous MacOS, pour que Siril détecte et utilise correctement StarNet, il est nécessaire de régler d'abord certains problèmes de permissions et de sécurité. Commencez par ouvrir l'application Terminal à partir du dossier Utilitaires

dans Applications. Dans le Terminal, vous devez changer votre répertoire de travail de votre répertoire personnel au répertoire d'installation de StarNetCLI. Pour ce faire, tapez `cd` suivi d'un `espace` puis faites glisser le dossier StarNetCLI dans la fenêtre du terminal pour copier son chemin. Appuyez sur `entrer`. Tapez ensuite les quatre commandes suivantes, en appuyant sur `entrer` après chacune d'elles :

```
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
```

Ensuite, lors de la première utilisation avec Siril, l'exécution de StarNet échouera avec un avertissement concernant libtensorflow. Annulez cet avertissement. Ouvrez les Préférences Système et sous Confidentialité et sécurité, cliquez sur le bouton *Ouvrir quand même* pour libtensorflow. Après cela, StarNet devrait s'exécuter correctement dans Siril.

Astuce

Sur MacOS, un nouvel exécutable Starnet optimisé pour la puce Apple Silicon a été publié sur le site : https://www.starnetastro.com/experimental/. Cette nouvelle version est beaucoup plus rapide que la précédente car elle utilise le nouveau PyTorch accéléré par MPS (https://developer.apple.com/metal/pytorch/). De plus, cette version contient des binaires signés, suivez les instructions du fichier README.txt

Cependant, il est toujours possible pour Siril d'exécuter des binaires externes et c'est ce que nous avons décidé d'implémenter à partir de Siril 1.2.0. Pour les paramètres, veuillez vous référer à la *page* des préférences. Elle explique comment indiquer à Siril où se trouve StarNet.

Avertissement

Il s'agit de l'emplacement de la version de StarNet en ligne de commande qui doit être fournie, et non de la version graphique.

Notez que StarNet a besoin d'images sous forme d'images TIFF, donc si Siril est compilé sans le support de libtiff, l'intégration de StarNet ne sera pas disponible.

Le but premier de StarNet est de supprimer toutes les étoiles des images afin d'appliquer un processus différent entre les étoiles et le reste de l'image. Cela permet généralement de contrôler le gonflement des étoiles pendant les différents étirements, mais c'est également très utile pour créer des images de comètes où la vitesse de suivi des comètes peut être sensiblement différente de celui des étoiles lointaines.

L'outil est très facile à utiliser et seules cinq options sont disponibles :

— **Pré-étirer l'image linéaire** : Si cette option est sélectionnée, un étirement optimisé de la fonction de transfert des tons moyens (MTF) est appliqué à l'image avant l'exécution de StarNet, et l'étirement inverse est appliqué à la fin. Ceci est nécessaire pour utiliser StarNet à l'étape linéaire du traitement.

— **Recomposer les étoiles à la fin de l'opération** : S'il est sélectionné, à la fin du processus de suppression des étoiles, l'outil de recomposition des étoiles s'ouvrira, fournissant une interface pour étirer et mélanger indépendamment l'arrière-plan et les étoiles si une réduction des étoiles, plutôt qu'une suppression totale, est souhaitée. Cette option n'a aucun effet lors du traitement d'une séquence.

— **Générer masque d'étoiles** : Cela va générer un masque d'étoiles et l'enregistrer dans le répertoire de travail. Le masque des étoiles est calculé comme la différence entre l'image originale et l'image sans étoiles. Le comportement par défaut est de produire un masque d'étoile.

— **Ré-échantillonner x2** : Cette option permet de sur-échantillonner l'image par un facteur 2 avant d'exécuter StarNet. Cela améliore les performances sur les étoiles très serrées mais quadruple le temps de traitement et peut nuire aux performances sur les très grandes étoiles. L'image est redimensionnée à la taille originale à la fin

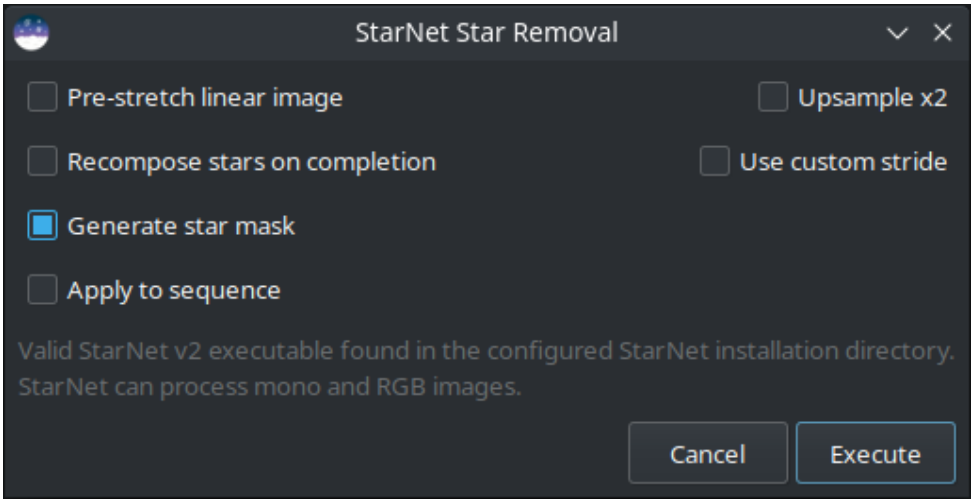


FIG. 66 – Boîte de dialogue StarNet.

de l'opération.

— **Utiliser un pas personnalisé** : Une valeur personnalisée peut être saisie pour le paramètre de pas de StarNet. La valeur par défaut est de 256 et le développeur de StarNet recommande de ne pas la modifier.

Le processus StarNet peut facilement être appliqué à une séquence. Le bouton *Appliquer à la séquence* permet de choisir si le processus sera appliqué à une seule image ou à une séquence. Lorsque le processus est appliqué à une séquence, une nouvelle séquence est créée contenant les images sans étoile et, si la génération de masque d'étoile est sélectionnée, une seconde séquence est créée contenant les images de masque d'étoile correspondantes.

Vous trouverez de plus amples informations sur StarNet sur le site web original (Anglais).

Un clic sur *Executer* lancera le processus. Cela peut être lent, en fonction des performances de votre machine. Cependant, Siril affiche une barre de progression pour suivre le traitement. Comme avec les autres processus Siril, si vous traitez une séquence, la barre de progression ne sera mise à jour qu'après l'achèvement de chaque image de la séquence, et montrera la progression globale de la séquence.

Commandes

Ligne de commande Siril

```
starnet [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Appelle StarNet pour supprimer les étoiles de l'image chargée.

StarNet est un programme externe, sans affiliation avec Siril, et doit être installé correctement avant la première utilisation de cette commande, avec le chemin d'accès à son installation en version CLI correctement défini dans Préférences / Divers.

L'image sans étoile (starless) est chargée à la fin et une image de masque d'étoile est créée dans le répertoire de travail à moins que le paramètre facultatif **-nostarmask** ne soit fourni.

En option, des paramètres peuvent être passés à la commande :
- 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

Ligne de commande Siril

```
seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Cette commande appelle StarNet++ pour supprimer les étoiles de la séquence **sequencename**. Voir STARNET

Liens : *starnet*

10.5.2 Recomposition des étoiles

L'outil de recomposition des étoiles est un outil graphique qui permet de combiner des images sans étoiles et des images avec masque d'étoiles. Il n'offre aucune manipulation d'image unique qui ne puisse être réalisée par d'autres moyens, par exemple en utilisant PixelMath et l'outil d'étirement hyperbolique généralisé, mais il offre un aperçu en temps réel de la combinaison de deux images distinctes auxquelles des étirements différents ont été appliqués à chacune d'elles.

Il n'y a pas d'équivalent en ligne de commande pour cet outil car il est purement graphique, mais les images sans étoile et avec masque d'étoile peuvent être combinées en utilisant les commandes *pm* et GHT (*ght*, *invght*, *modasinh*, *invmodasinh* et *linstretch*).

L'outil se trouve dans le menu Traitement des images, dans le sous-menu Traitement des étoiles.

La boîte de dialogue est divisée en deux colonnes, une pour chacune des images d'entrée.

Chaque image d'entrée est chargée à l'aide du sélecteur de fichiers correspondant. Chaque colonne présente un aperçu de l'histogramme, qui peut être réduit pour faciliter l'utilisation sur les petits écrans, ainsi qu'un ensemble de commandes d'étirement.

Le mode d'histogramme peut être changé entre linéaire et logarithmique en utilisant la case à cocher au bas de la boîte de dialogue. Cette boîte de dialogue respecte la préférence globale de Siril concernant les histogrammes linéaires ou logarithmiques, qui peut être définie dans la fenêtre des Préférences.

Astuce

Star Recomposition does not obey the FITS 16-bit preference as there could be an adverse effect on performance in some cases (the calculations must be done using floating point maths anyway). So if you really want a 16-bit result to be created from 32-bit input images then you will need to use the *precision switch*.

Mode Simple

La boîte de dialogue a deux vues, qui déterminent les commandes affichées. Elle s'ouvre en mode simple, qui n'affiche que les commandes les plus utiles pour une combinaison typique de starless et de starmask.

— Le type d'étirement de l'image sans étoile est défini sur l'étirement hyperbolique généralisé et les contrôles Facteur d'étirement, Intensité d'étirement local, Point de symétrie et Point noir sont affichés. Outre le contrôle

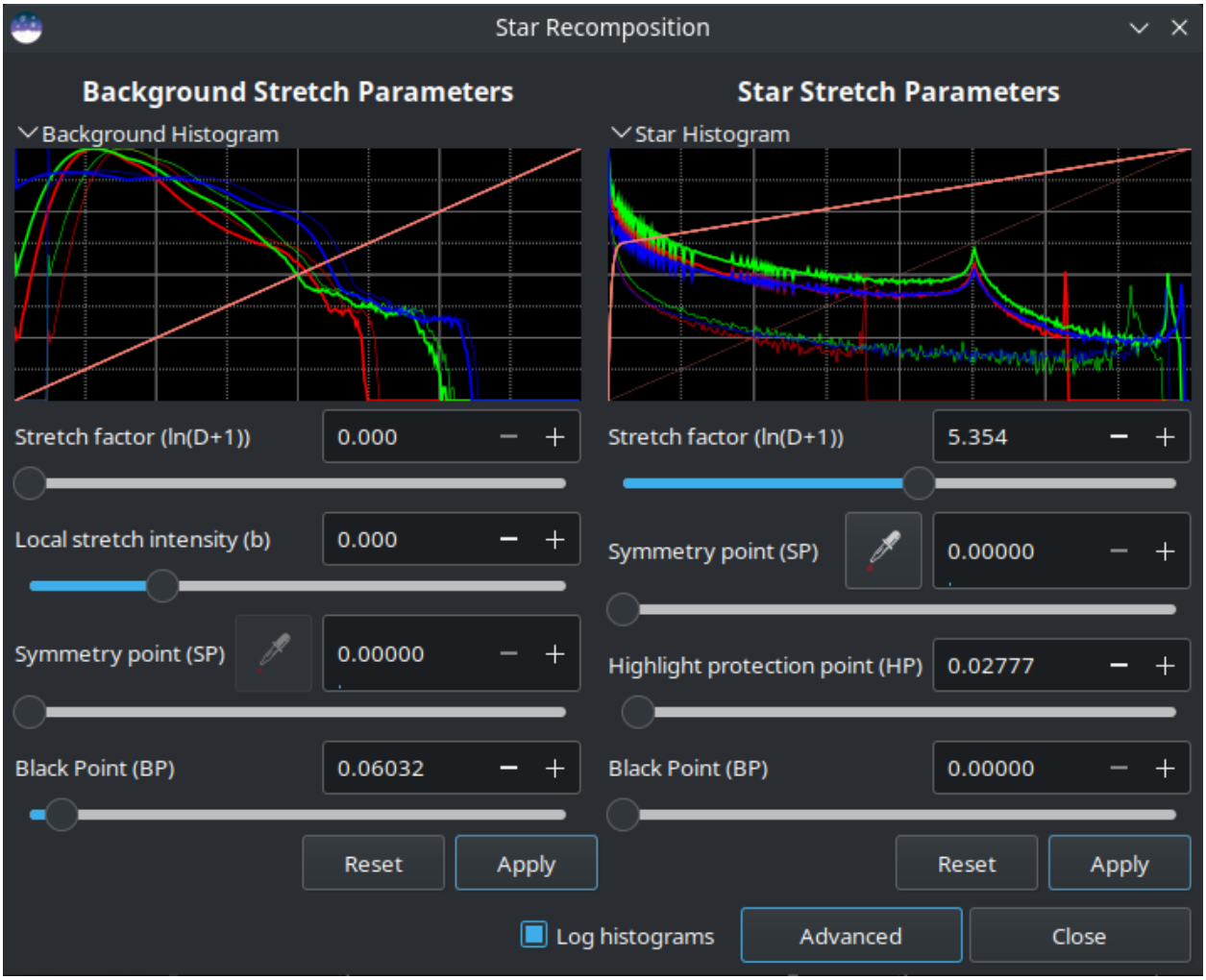


FIG. 67 – Boîte de dialogue Recomposition de l'étoile.

du point de symétrie dédié, il peut être défini à l'aide de la pipette pour sélectionner la valeur moyenne des pixels d'une sélection de l'image.
Notez que la pipette est désactivée lorsqu'un décalage du point noir n'a pas été appliqué : en raison du processus d'application de l'étirement hyperbolique puis du décalage de point noir, le comportement de l'outil devient non intuitif lorsqu'un paramètre du point noir non nul est défini. Pour résoudre ce problème, il suffit d'appliquer le décalage du point noir et la pipette redeviendra disponible pour votre prochain étirement hyperbolique.

— Le type d'étirement de l'image du masque stellaire est défini sur Modified Arcsinh stretch et les contrôles Stretch Factor et Highlight Protection sont affichés.
— Le modèle de couleur de la luminosité qui simule la vision humaine est utilisé pour les deux séries d'étirements : il permet de mieux préserver les couleurs dans l'image non étirée.

Les détails de toutes les commandes d'étirement, aussi bien celles affichées en mode simple que celles affichées en mode avancé, sont disponibles sur la page de documentation Étirement hyperbolique généralisé.

Le contrôle du point noir (BP) fonctionne d'une manière légèrement différente du contrôle BP dans l'outil autonome d'étirement linéaire (ajustement du point noir) de l'étirement hyperbolique généralisé (GHS). Dans cet outil, l'ajustement du point noir est appliqué *après* l'étirement hyperbolique, alors que dans l'outil autonome, il s'agit d'un étirement séparé appliqué seul. En essayant d'optimiser la combinaison d'étirements indépendants sur les deux images d'entrée, cette approche s'est avérée la plus pratique. Cela signifie que la quantité de décalage du point noir requise dans cet outil est différente de celle requise dans l'outil GHS, et que le point noir ne peut pas être défini en cliquant sur l'histogramme.

Chaque étirement est indépendant. Les paramètres d'étirement pour le côté sans étoile peuvent être appliqués à l'aide du bouton Appliquer de gauche : l'image sans étoile est étirée en fonction des paramètres d'étirement actuels, puis les paramètres d'étirement sont réinitialisés afin que d'autres étirements puissent être appliqués de manière itérative. De même, les paramètres d'étirement du masque étoilé peuvent être appliqués à l'aide du bouton Appliquer situé à droite. Chaque ensemble de paramètres d'étirement peut être réinitialisé aux valeurs par défaut à l'aide de son bouton Réinitialiser respectif.

La boîte de dialogue peut être basculée entre le mode simple et le mode avancé à l'aide du bouton situé en bas de l'écran.

Mode Avancé

En mode avancé, la gamme complète des contrôles d'étirement de l'outil d'étirement hyperbolique généralisé est disponible, y compris le type d'étirement, le modèle d'étirement des couleurs et le point de protection des ombres pour les deux images d'entrée. Cela permet de personnaliser davantage les deux étirements si nécessaire. Si l'interface utilisateur est ramenée en mode simple, toutes les modifications effectuées à l'aide des commandes avancées restent effectives, seules les commandes sont masquées.

> **Note**
>
> Il n'est pas possible d'étirer les canaux de saturation dans cet outil. L'outil est déjà assez gourmand en mémoire et en CPU : doubler la mémoire requise en ajoutant une copie TSL de chaque image de travail est considéré comme excessif. La saturation peut facilement être étirée séparément une fois la combinaison terminée.

10.5.3 Désaturer les étoiles

Lorsqu'une recherche d'étoiles est appliqué à une image (dont les données sont toujours linéaires), des ellipses sont affichées autour des étoiles. Lorsqu'une ellipse est magenta, cela signifie que l'étoile est saturée.

Une étoile saturée est une étoile dont les pixels les plus lumineux n'ont plus d'information et sont coupés à la valeur maximale. En général, nous essayons de ne pas saturer les étoiles, même si cela n'est pas possible pour les plus brillantes. Si malgré les précautions prises il reste des étoiles saturées, Siril dispose d'un algorithme qui va reconstruire le profil de l'étoile en tenant compte des résultats de l'ajustement effectué lors de la recherche de l'étoile.

Tout d'abord, vous devez effectuer une détection d'étoile, soit avec la commande *findstar* soit avec le bouton de la

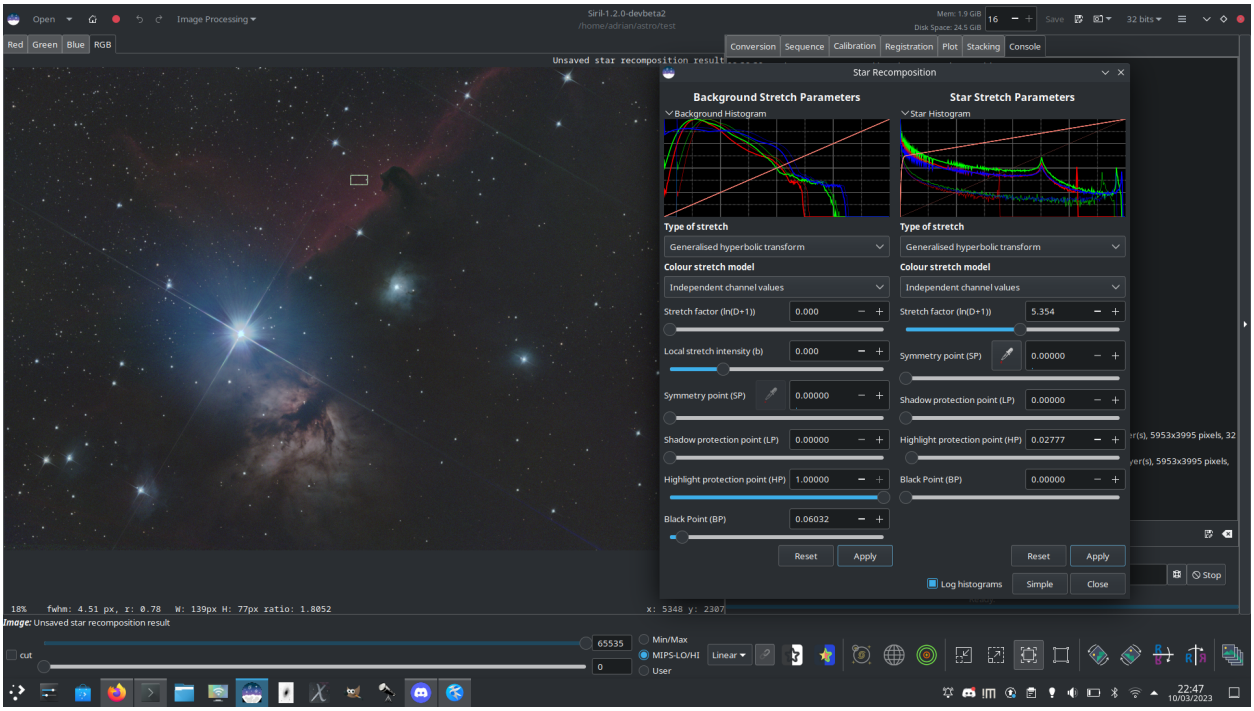


Fig. 68 – La recomposition des étoiles permet de combiner une image sans étoile et un masque d'étoiles de la région d'Alnitak

fenêtre *PSF Dynamique*. Ensuite, l'outil de désaturation se trouve dans *Traitement des étoiles* → *Désaturer les étoiles*.

Astuce

Nous recommandons d'utiliser un profil Moffat dans la fenêtre *PSF Dynamique* pour obtenir de meilleurs paramètres.

Avertissement

Il est important d'utiliser l'outil sur une image linéaire, autrement les étoiles n'auront pas un profil Gaussien/Moffat et les calculs seront invalides.

Après avoir cliqué sur l'outil, Siril passe à l'onglet console et affiche les résultats du processus en cours :

```
22:26:17: Star synthesis (desaturating clipped star profiles): processing...
22:26:17: Findstar: processing for channel 0...
22:26:21: Star synthesis: desaturating stars in channel 0...
22:26:21: Star synthesis: 70 stars desaturated
22:26:21: Remapping output to floating point range 0.0 to 1.0
22:26:21: Execution time: 4.09 s
```

Il est nécessaire de refaire une détection d'étoiles pour voir les changements.

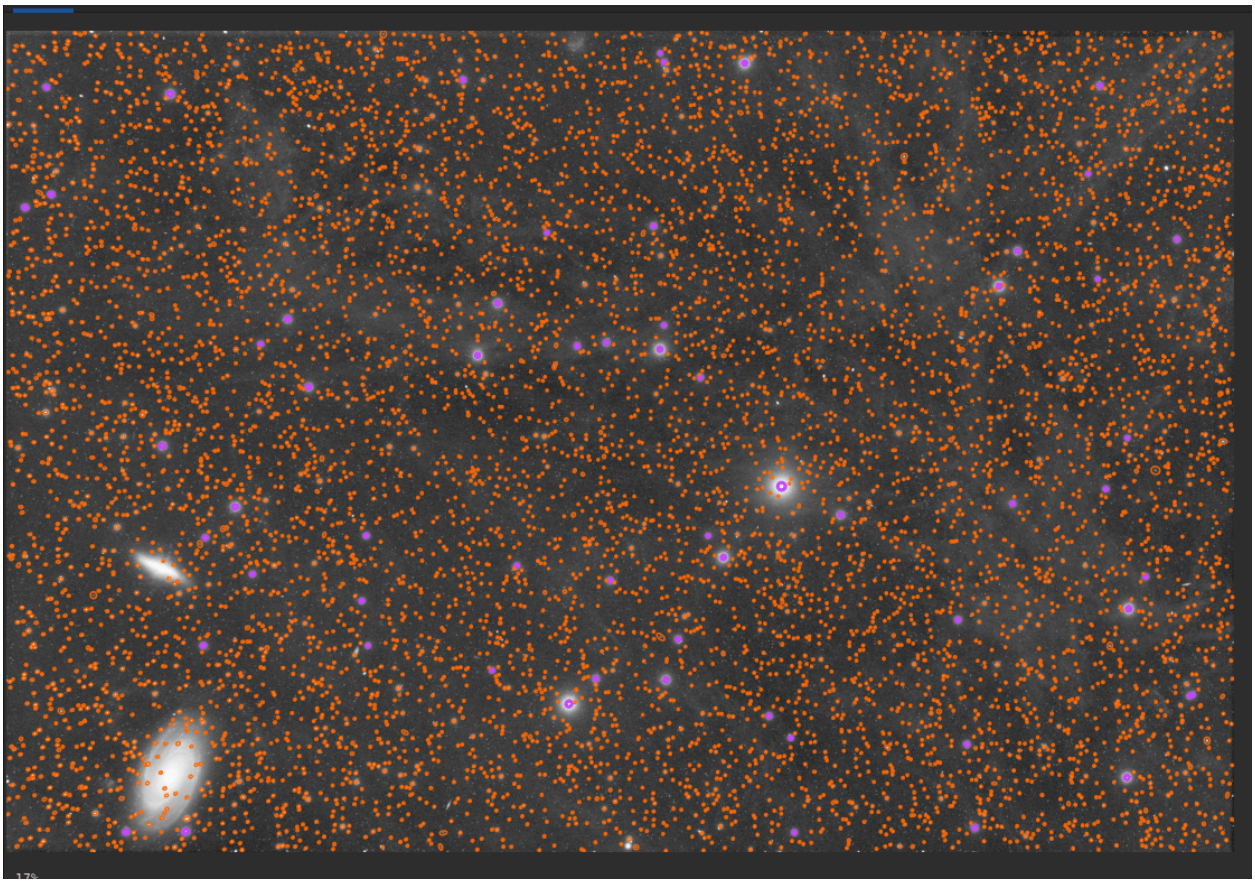


FIG. 69 – Une détection d'étoile montre toutes les étoiles trouvées par Siril. Les ellipses magenta correspondent aux étoiles saturées. L'image est affichée en mode "auto-étirée" : les données sont toujours linéaires.

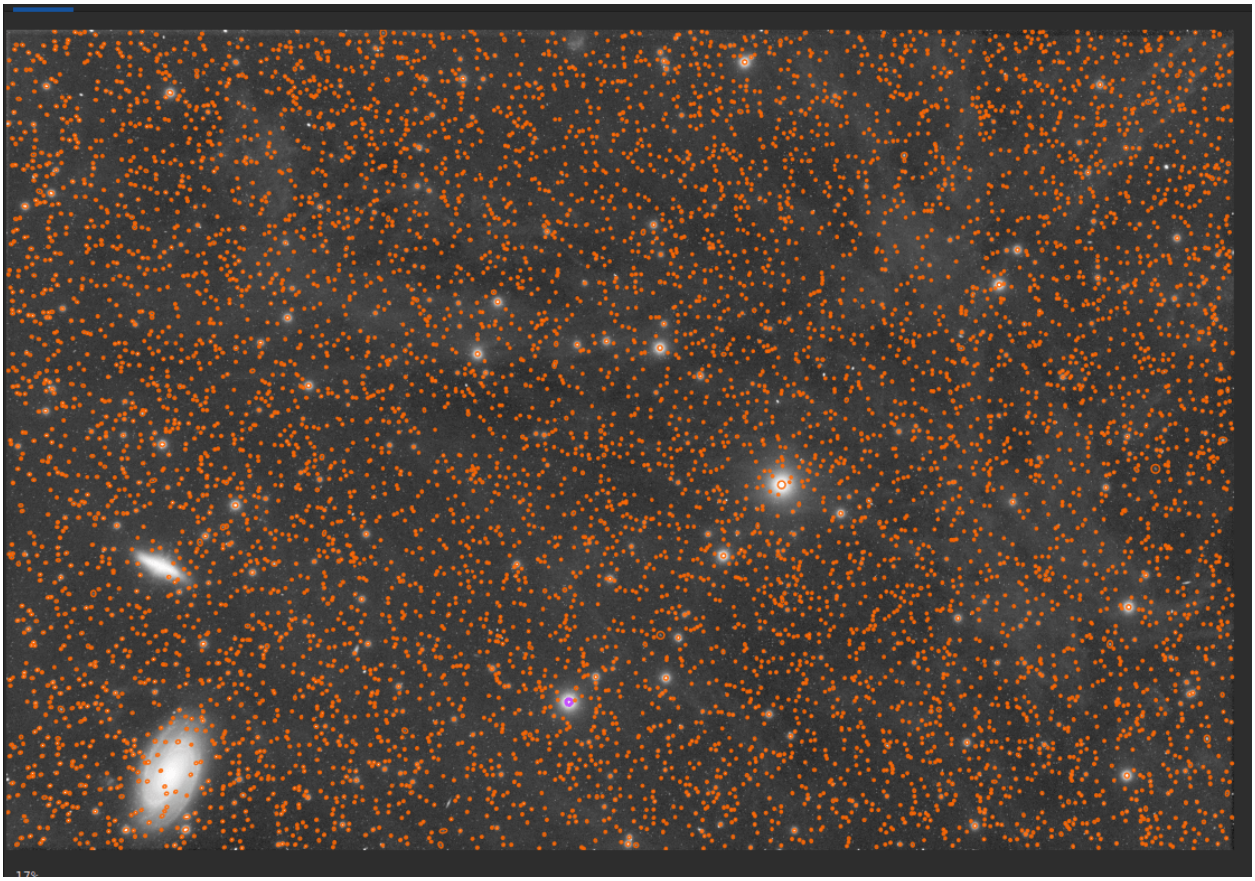


FIG. 70 – Après un traitement de désaturation, plus aucune ellipse magenta n'est visible. Toutes les étoiles ont été reconstruites. L'image est affichée en vue auto-étirement : les données sont toujours linéaires.

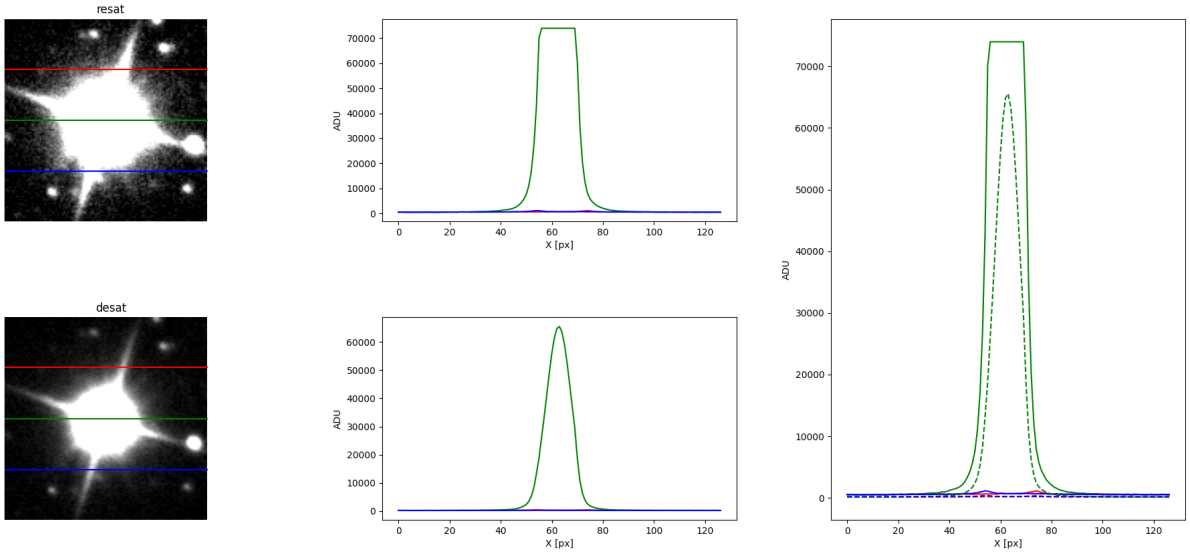


FIG. 71 – Comparaison d'un étoile avant et après utilisation de l'outil de désaturation.

Ligne de commande Siril

```
unclipstars
```

Reprofile les étoiles coupées de l'image chargée pour les désaturer, en mettant à l'échelle la sortie de manière à ce que toutes les valeurs de pixels soient <= 1,0

10.5.4 Resynthèse intégrale

L'outil Resynthèse intégrale a pour but d'aider à corriger les étoiles fortement déformées en utilisant les fonctions d'ajustement stellaire de Siril. Il peut être utile pour sauver des images qui souffrent de coma ou d'autres distorsions. Si Siril peut détecter les étoiles, il peut les corriger.

L'outil se trouve dans le menu Traitement des images, dans le sous-menu Traitement des étoiles.

Le résultat de l'outil est un masque stellaire synthétique. Pour l'utiliser, il doit être recombiné avec une version sans étoiles de l'image originale. Celle-ci peut être préparée à l'aide de la commande *starnet* ou de l'outil Starnet GUI, ou encore à l'aide d'un logiciel tiers de suppression des étoiles.

Cet outil n'a pas d'options, il suffit de cliquer sur l'élément de menu pour l'utiliser, ou d'utiliser la commande *synthstar*.

Si aucune étoile n'a été détectée dans l'image, l'outil détectera automatiquement les étoiles en utilisant les paramètres actuels de modélisation des étoiles accessibles via l'outil PSF dynamique ou en utilisant la commande *setfindstar*.

Si les étoiles ont été modélisées à l'aide de l'outil PSF dynamique ou de la commande *findstar*, les étoiles détectées seront resynthétisées en utilisant leurs profils de luminosité individuels modélisés. Un raccourci vers l'outil PSF dynamique est fourni au moyen du bouton de configuration dans le menu de l'interface graphique à côté de l'outil Resynthèse complète.

Il est recommandé de procéder d'abord à la détection manuelle des étoiles, car cela permet de vérifier les résultats : si des galaxies ont été incorrectement détectées comme des étoiles, elles peuvent être supprimées de la liste des étoiles avant de lancer la resynthèse.

Une fois le masque stellaire synthétique créé, il peut être combiné avec l'image sans étoile à l'aide de l'outil de recombinaison stellaire.

Commandes

Ligne de commande Siril

```
synthstar
```

Corrige les étoiles imparfaites de l'image chargée. Quelle que soit l'importance de la coma, de la dérive ou de toute autre distorsion de vos étoiles, si la routine de recherche d'étoiles de Siril peut les détecter, synthstar les corrigera. Pour une utilisation intensive, vous pouvez souhaiter détecter manuellement toutes les étoiles que vous souhaitez corriger. Cela peut être fait en utilisant la commande findstar de la console ou la boîte de dialogue PSF dynamique. Si vous n'avez pas lancé la détection d'étoiles, elle sera lancée automatiquement avec les paramètres par défaut.

Pour de meilleurs résultats, synthstar doit être exécuté avant l'étirement.

La sortie de synthstar est un masque d'étoile synthétique entièrement corrigé comprenant des PSF d'étoiles parfaitement rondes (profils Moffat ou gaussiens en fonction de la saturation de l'étoile) calculés pour correspondre à l'intensité, à la FWHM, à la teinte et à la saturation mesurées pour chaque étoile détectée dans l'image d'entrée. Cela peut ensuite être recombiné avec l'image sans étoiles pour produire une image avec des étoiles parfaites.

Aucun paramètre n'est requis pour cette commande

Liens : *psf*

10.6 Géométrie

10.6.1 Rotation

Rotation de 90 degrés

Il est possible de faire pivoter l'image de 90 degrés dans le sens des aiguilles d'une montre et dans le sens inverse des aiguilles d'une montre à l'aide du menu dédié. Ici, la rotation est effectuée sans interpolation des pixels et c'est donc la méthode préférée si vous souhaitez faire pivoter l'image d'un multiple de 90 degrés. Cette fonction est également accessible via les icônes et dans la barre d'outils.

Rotation&Recadrage

Pour une rotation d'un autre angle, vous devez utiliser l'outil Rotation&Recadrage. Il permet une rotation et un recadrage précis qui peuvent être facilement contrôlés.

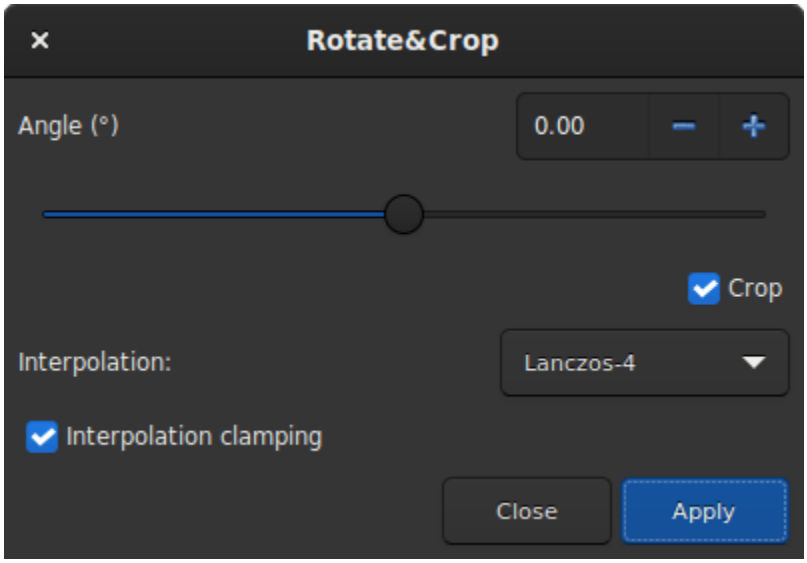


Fig. 72 – Boite de dialogue Rotation&Recadrage affichant tous les paramètres.

Cinq algorithmes d'interpolation sont disponibles :
- Plus Proches Voisins
- Bilinéaire
- Bicubique
- Zone de relation des pixels

— Lanczos-4 (Défaut)

Lanczos-4 est celui qui donne les meilleurs résultats. Cependant, si vous voyez des artefacts, en particulier des étoiles entourées de pixels noirs, vous pouvez essayer d'autres méthodes. Cependant, le bouton *Contrainte de l'interpolation* applique un facteur de contrainte à l'interpolation Bicubique et Lanczos-4 afin d'éviter les artefacts en annaux.

Si vous ne voulez pas que l'image soit recadrée après la rotation, vous devez décocher le bouton *recadrer*. Cependant, les zones manquantes de l'image seront remplies de pixels noirs.

L'intérêt de cet outil est que la rotation de l'image est représentée par un cadre rouge, comme illustré dans la figure ci-dessous. De plus, si une sélection est active, il est possible de changer sa taille et de voir en temps réel l'évolution du cadrage.

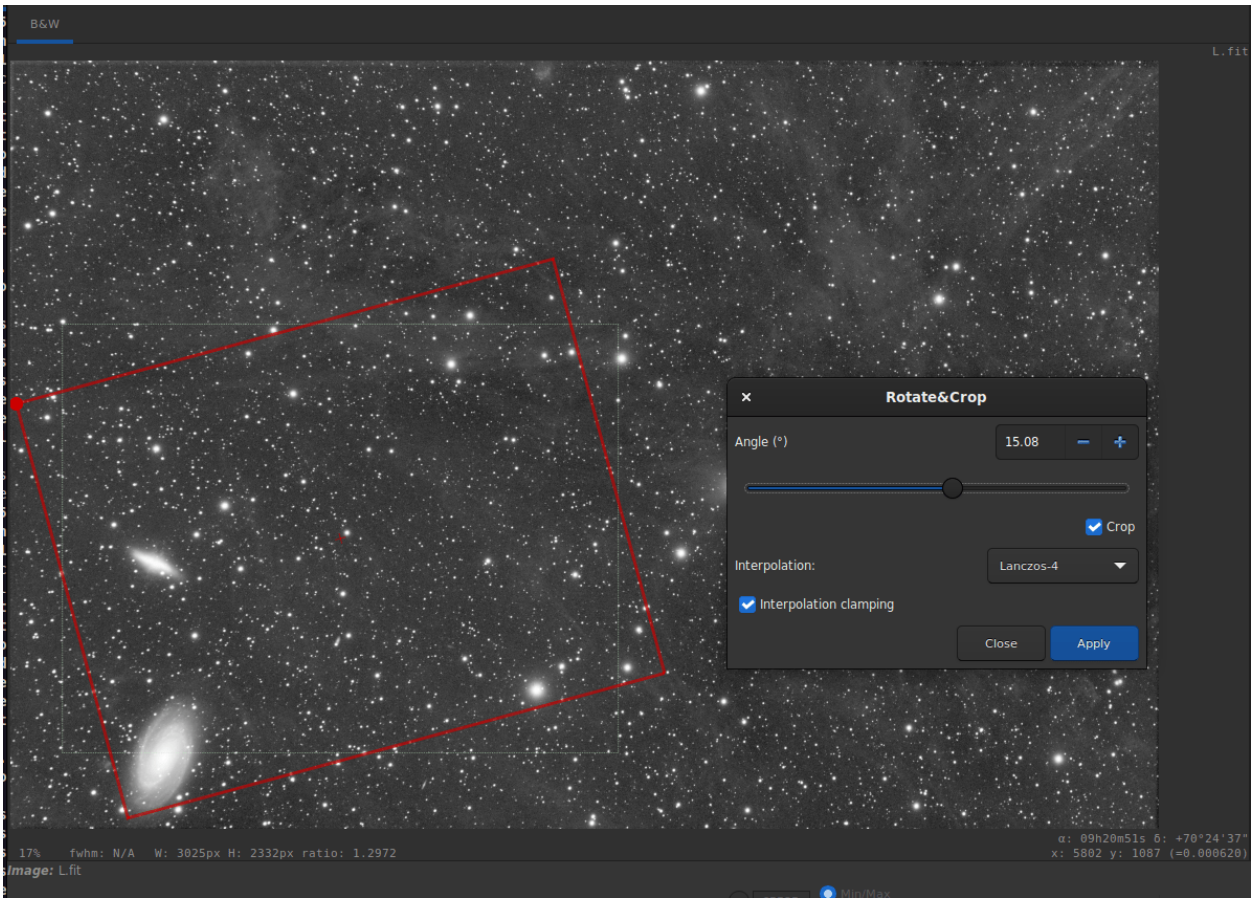


Fig. 73 – Boîte de dialogue Rotation et recadrage avec une sélection active. Cliquez pour agrandir la figure et mieux voir les détails.

Ligne de commande Siril

```
rotatePi
```

Fait pivoter l'image chargée d'un angle de 180° autour de son centre. Ceci est équivalent à la commande "ROTATE 180" ou "ROTATE -180"

Liens : *rotate*

Ligne de commande Siril

```
rotate degree [-nocrop] [-interp=] [-noclamp]
```

Fait pivoter l'image chargée d'un angle de **degree** degrés. L'option **-nocrop** peut être ajoutée pour éviter le recadrage à la taille de l'image (des bordures noires seront ajoutées).

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.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.
La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**

10.6.2 Miroir

Il est également possible d'appliquer une transformation miroir à l'image. Soit le long de l'axe x, soit le long de l'axe y.

Cette transformation est également accessible via les boutons et de la barre d'outils.

Ligne de commande Siril

```
mirrorx [-bottomup]
```

Retourne l'image chargée autour de l'axe horizontal. L'option **-bottomup** ne retournera l'image que si elle n'est pas bottom-up

Ligne de commande Siril

```
mirrory
```

Retourne l'image autour de l'axe vertical

10.6.3 Binning

Le binning est une transformation spéciale pour le rééchantillonnage d'une image. Il calcule la somme ou la moyenne des pixels 2x2, 3x3, ... (en fonction du facteur de binning) de l'image en mémoire (comme le binning analogique de la caméra CCD).

Ligne de commande Siril

```
binxy coefficient [-sum]
```

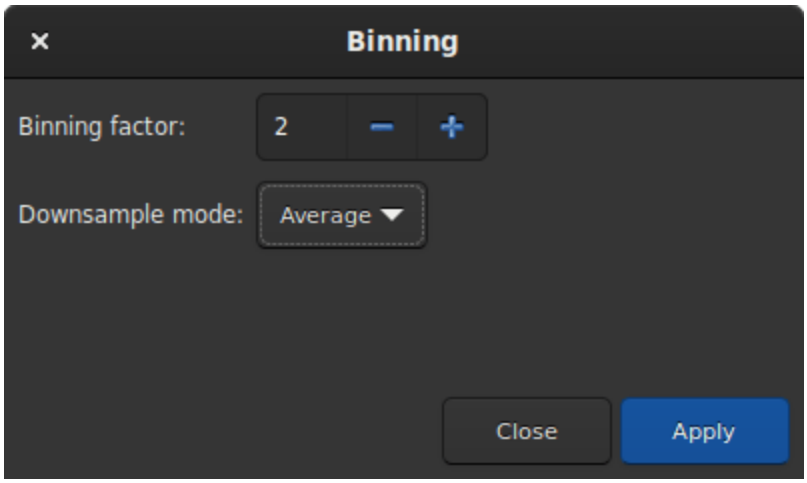


FIG. 74 – Boite de dialogue binning

Calcule le binning numérique de l'image en mémoire (somme des pixels 2x2, 3x3..., comme le binning analogique des caméras CCD). Si l'argument facultatif **-sum** est passé, alors la somme des pixels est calculée, tandis que c'est la moyenne lorsque aucun argument facultatif n'est fourni

10.6.4 Ré-échantillonner

L'outil de rééchantillonnage permet de redimensionner l'image au prix d'une interpolation choisie dans la liste suivante :

- Plus Proches Voisins
- Bilinéaire
- Bicubique
- Zone de relation des pixels
- Lanczos-4 (Défaut)

Lanczos-4 est celui qui donne les meilleurs résultats. Cependant, si vous voyez des artefacts, en particulier des étoiles entourées de pixels noirs, vous pouvez essayer d'autres méthodes. Cependant, le bouton *Contrainte de l'interpolation* applique un facteur de contrainte à l'interpolation Bicubique et Lanczos-4 afin d'éviter les artefacts en annaux.

Si vous souhaitez modifier le rapport d'image, vous devez décocher le bouton *Préserver le ratio*.

Ligne de commande Siril

```
resample { factor | -width= | -height= | -maxdim= } [-interp=] [-noclamp]
```

Resamples the loaded image, either with a factor **factor** or for the target width or height provided by either of **-width=**, **-height=** or **-maxdim=**. This is generally used to resize images : a factor of 0.5 divides size by 2. The **-maxdim** argument can be used to resize the longest dimension of the image to a set size, which can be useful for optimizing images for certain websites, e.g. social media websites.
Dans l'interface graphique utilisateur, nous pouvons voir que plusieurs algorithmes d'interpolation sont proposés.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi d'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}.
La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**

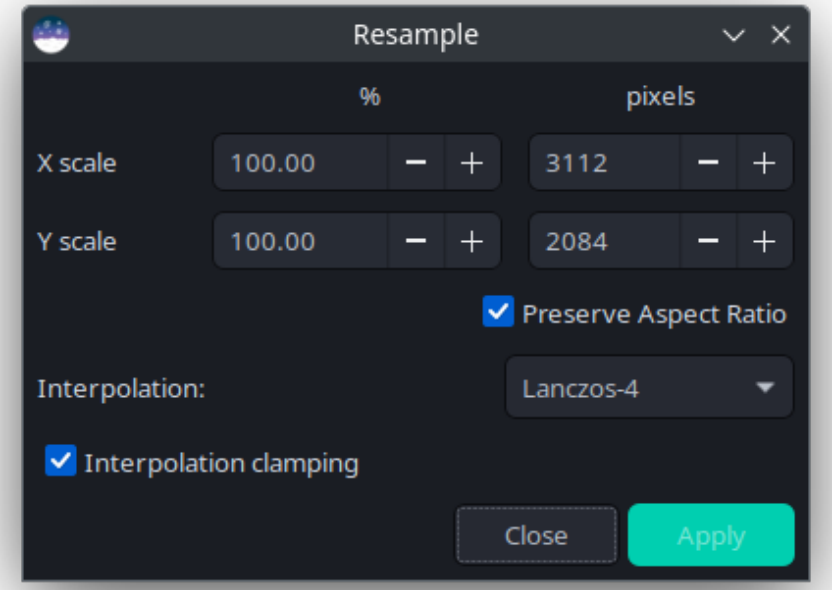


FIG. 75 – Boite de dialogue Ré-échantillonner

```
seqresample sequencename { -scale= | -width= | -height= } [-interp=] [-prefix=]
```

Met à l'échelle de la séquence donnée en argument **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

The scale factor is specified either by the **-scale=** argument or by setting the output width, height or maximum dimension using the **-width=**, **-height=** or **-maxdim=** options.

Une méthode d'interpolation peut être spécifiée en utilisant l'argument **-interp=** suivi d'une des méthodes de la liste **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}... Le clamping est appliqué pour l'interpolation cubique et lanczos.

Le nom de la séquence de sortie commence par le préfixe "scaled_", sauf indication contraire avec l'option **-prefix=**

10.7 Extraction de gradient

Le fond du ciel présente souvent un gradient indésirable causé par la pollution lumineuse, la lune ou simplement l'orientation de la caméra par rapport au sol. Cette fonction échantillonne le fond en de nombreux endroits de l'image et recherche une tendance dans les variations. Elle le supprime en suivant une fonction lissée pour éviter de supprimer des nébuleuses.

Les échantillons peuvent être placés automatiquement en fournissant une densité (*Echantillons par ligne*) et en cliquant sur *Générer*. Si des zones de l'image sont plus lumineuses que la médiane d'un facteur *Tolérance de la grille* multiplié par sigma, alors aucun échantillon ne sera placé à cet endroit.

Astuce

Si vous avez des gradients très forts, par exemple lorsque vous photographiez dans un ciel urbain à fort Bortle, même la tolérance maximale de la grille peut s'avérer insuffisante. Dans ce cas, vous pouvez cocher la case *Gardez tous les échantillons* et la grille d'échantillons complète sera remplie. Vous devrez alors retirer manuellement les échantillons posés sur des objets astronomiques réels.

Après la génération, les échantillons peuvent également être ajoutés manuellement (clic gauche) ou retirés manuellement (clic droit).

Il y a deux algorithmes pour supprimer le gradient :

10.7.1 RBF

Il s'agit de la méthode la plus moderne. Elle utilise la fonction de base radiale pour synthétiser un fond de ciel afin de supprimer le gradient avec une grande flexibilité. Elle nécessite un seul paramètre qui est présent sous la forme d'un curseur :*Lissage*. Cette valeur permet de déterminer la douceur ou la dureté de la transition entre les points d'échantillonnage. Un facteur de lissage élevé a du sens pour les gradients larges et uniformes, et une valeur correspondante plus faible pour de petits gradients locaux.

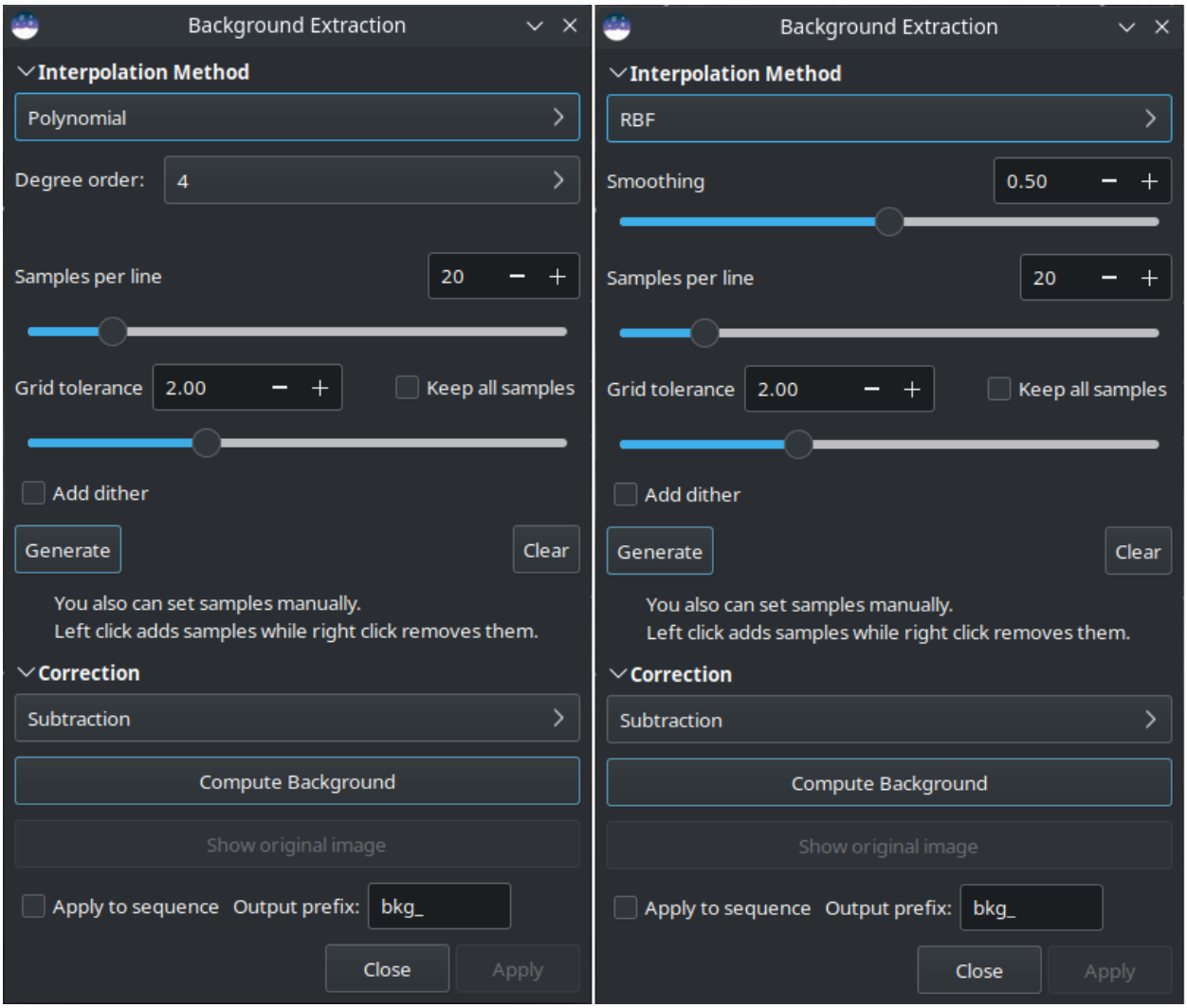


Fig. 76 – Boite de dialogue extraction du gradient. Sur la gauche la version polynomiale, sur la droite la version RBF.

Astuce

Commencez par le réglage de base (50 %) et ajustez progressivement pour obtenir des résultats optimaux.

Théorie

Les fonctions de base radiales sont des fonctions de la forme$\phi(\mathbf{x}) = \phi(\|\mathbf{x}\|)$, sachant que dans notre cas, nous utilisons la norme euclidienne$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2}$. La fonction$f$, qui décrit le modèle d'arrière-plan, peut maintenant être exprimée comme une combinaison linéaire

$$f(\mathbf{x}) = \sum_i w_i \, \phi(\|\mathbf{x} - \mathbf{x_i}\|) + o$$

oùw_i correspond aux poids des différents points d'échantillonnage et$o$ correspond à un décalage constant.

L'exigence selon laquelle la fonction$f$ doit passer par les points d'échantillonnage se traduit par la condition suivante

$$\begin{pmatrix} \phi(\mathbf{x}_1 - \mathbf{x}_1) & \phi(\mathbf{x}_1 - \mathbf{x}_2) & \dots & \phi(\mathbf{x}_1 - \mathbf{x}_N) & 1 \\ \phi(\mathbf{x}_2 - \mathbf{x}_1) & \phi(\mathbf{x}_2 - \mathbf{x}_2) & \dots & \phi(\mathbf{x}_2 - \mathbf{x}_N) & 1 \\ \vdots & \vdots & \ddots & \vdots & \\ \phi(\mathbf{x}_N - \mathbf{x}_1) & \phi(\mathbf{x}_N - \mathbf{x}_2) & \dots & \phi(\mathbf{x}_N - \mathbf{x}_N) & 1 \\ 1 & 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \\ o \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ 0 \end{pmatrix},$$

qui ne peut être satisfaite que si la matrice du côté gauche est inversible. Avec le bon choix de la fonctionϕ, cela peut toujours être garanti [Wright2003].

En outre, la somme$s\,I$ est ajoutée à la matrice du côté gauche, oùs est un paramètre de lissage et$I$ est la matrice unitaire. La somme entraîne une régularisation, qui donne un résultat d'autant plus lisse que le paramètre$s$ est grand. Ce paramètre peut être modifié avec le paramètre*Lissage* de la boîte de dialogue.

Pour la fonction de base radiale, nous utilisons la spline mince$\phi(|\mathbf{x}|) = |\mathbf{x}|^2 \log(|\mathbf{x}|)$.

10.7.2 Polynomiale

Il s'agit de l'algorithme original et le plus simple développé dans Siril. Un seul paramètre est utilisé dans le calcul polynomial : le bouton*Ordre*. Plus le degré de l'ordre est élevé, plus la correction est souple, mais un degré trop élevé peut donner des résultats étranges comme une surcorrection.

Astuce

Une correction de degré 1 peut s'avérer très utile lorsque vous souhaitez supprimer le gradient sur les images individuelles.

Avertissement

Background removal can be carried out on CFA images, but only if they have Bayer patterns. (It is not supported for X-TRANS patterns.) For Bayer patterned images, the image is treated as four spatially interleaved images, each corresponding to a CFA subchannel. Each subchannel is independently processed to remove its gradient and then the subchannels are recombined into the original interleaved pattern.

The intended use for this is to remove linear gradients from sequence frames prior to using Drizzle on a Bayer patterned sequence, and in that case it is strongly recommended to use linear (polynomial order 1) gradient removal as with other pre-stacking gradient removal.

10.7.3 Paramètres généraux

- **Ajout de diffusion d'erreur (dither)** : Activez cette option lorsque des bandes de couleur apparaissent après l'extraction du gradient. Le dither est une forme de bruit appliquée intentionnellement et utilisée pour randomiser l'erreur de quantification, empêchant ainsi l'apparition de motifs à grande échelle tels que les bandes de couleur dans les images.
- **Correction** :
  - **Soustraction** : ceci est principalement utilisé pour corriger les effets additifs, tels que les gradients causés par la pollution lumineuse ou par la lune.
  - **Division** : principalement utilisé pour corriger les phénomènes multiplicatifs tels que le vignettage ou l'absorption atmosphérique différentielle par exemple. Cependant, ce genre d'opération devrait être effectué à l'aide du master-flat.
- **Calcul du fond de ciel** : Cela va calculer le fond synthétique et appliquer la correction sélectionnée. Le modèle est toujours calculé à partir de l'image d'origine conservée en mémoire, ce qui permet à l'utilisateur de travailler de manière itérative.
- **Afficher l'image originale** : Appuyez sur ce bouton pour voir l'image originale.

Le gradient du fond de l'image pré-traitée peut être complexe car le gradient peut avoir tourné avec la session d'acquisition. Il peut être difficile de le supprimer complètement car il est difficile de le représenter par une fonction polynomiale. Si c'est le cas, vous pouvez envisager de supprimer le gradient dans les images individuelles : dans une seule image, le gradient de fond est beaucoup plus simple et suit généralement une fonction linéaire simple (degré 1).

Astuce

Il arrive que des bandes de couleur disgracieuses apparaissent après l'extraction du gradient. Dans ce cas, il y a deux choses à vérifier. Tout d'abord, si l'image est en 16 bits, nous vous conseillons vivement de toujours utiliser le format 32 bits. Si, malgré tout, vous observez de tels artefacts, l'option **Ajout de diffusion d'erreur (dither)**, expliquée ci-dessus, est la solution à votre problème.

FIG. 77 – Lorsqu'un tel effet de bande se produit après l'extraction du gradient, il peut être résolu par l'option **Ajout de diffusion d'erreur (dither)** (avec l'aimable autorisation de Nathan B.).

Astuce

Pour obtenir de bons résultats avec l'algorithme RBF, il suffit généralement de moins d'échantillons qu'avec l'algorithme polynomial.

Voir aussi

Pour plus d'explications, voir le tutoriel correspondant ici.

Ligne de commande Siril

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-
↪existing]
```

Calcule un fond de ciel synthétique en utilisant soit le modèle de fonction polynomiale de **degree** degrés, soit le modèle RBF (si **-rbf** est fourni à la place) et le soustrait de l'image.

Le nombre d'échantillons par ligne horizontale et la tolérance pour exclure les zones plus claires peuvent être

ajustés avec les arguments optionnels. La tolérance est exprimée en unités MAD : médiane + tolérance * mad.
La diffusion d'erreur, nécessaire pour les gradients peu dynamiques, peut être activé avec **-dither**.
Pour RBF, le paramètre de lissage supplémentaire est également disponible. Pour utiliser des échantillons d'arrière-plan préexistants (par exemple, si vous avez défini des échantillons d'arrière-plan à l'aide d'un script Python), l'argument **-existing** doit être utilisé

Astuce

The -existing command argument forces use of existing background samples. This option is primarily for use in conjunction with the Python module where `SirilInterface.set_bgsamples()` can be used to set custom background samples based on user-defined algorithms. If it is not provided, *subsky* will automatically regenerate background samples. Note that the -existing option is not available with the seqsubsky command, because sequence frames are not necessarly registered at the time background subtraction is carried out, so the samples for one frame do not necessarily apply to another.

Ligne de commande Siril

```
seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-
↪smooth=0.5] [-prefix=]
```

Même commande que SUBSKY mais pour la séquence **sequencename**.
Dithering, required for low dynamic gradients, can be disabled with **-nodither**. Note that the **-existing** option is not available for sequence background removal, as the frames of a sequence are not necessarily always aligned.

Le nom de la séquence de sortie commence par le préfixe "bkg_" sauf indication contraire avec l'option **-prefix=**.
Seules les images sélectionnées dans la séquence sont traitées

Liens : *subsky*

10.8 Interface GraXpert

Astuce

The first GraXpert interface was written in C and simply called the GraXpert executable. However for various reasons this approach proved unreliable for a significant minority of users and with Siril 1.4.0-beta2 an updated interface has been released as a Python script. This directly interfaces with the GraXpert model files and provides much better performance and integration, and since it builds on the way the Python interface uses the native PyPI modules library to install the correct versions of dependecies for each user's machine it is expected to be more reliable and better able to make use of different types of GPU.

Owing to the way GraXpert distributes its AI models from an Amazon S3 instance, a copy of the GraXpert program needs to be available in order to download model files. The model files you can download are dependent on the version of GraXpert installed therefore it is recommended to install 3.1.0-RC2 which provides access to the stellar and background object deconvolution models as well as the denoising and background extraction models that have been available since version 3.0.0.

The old GraXpert interface was marked as deprecated in 1.4.0-beta2 and has since been removed from the code base. This documentation covers the new Python-based interface.

Siril provides an interface to the AI models provided by the third party GraXpert tool. This is a third party piece of Free and Open Source software that provides sophisticated background removal and denoising routines. There is a lengthy history of collaboration between Siril and GraXpert, and the RBF algorithm used in Siril's core background removal function was contributed by the GraXpert developers several years ago.

In order to be used, the path to the working GraXpert installation must be set in *Preferences->Miscellaneous*.

10.8.1 Installation et configuration

The GraXpert-AI.py script can be installed in the usual way, by going to the *Scripts* menu -> *Get Scripts* and selecting the check box next to the script in the list.

To configure the script you need to tell Siril where your GraXpert installation is. To do this, go to *Preferences* -> *Miscellaneous* and select the GraXpert executable using the appropriate file chooser widget.

10.8.2 Aspects généraux du GUI

L'interface graphique est présentée ci-dessous :

The title bar changes to show the currently selected operation, and at the top of the window are the script title and version number. A dropdown allows you to select the operation you wish to perform. How this dropdown is populated depends on the version of GraXpert you have available. It will always feature Denoising and Background Extraction, even if no locla models are available (see the section on Model Manager for information on how to download models). If you have locally available deconvolution models, or if you have set the location of a GraXpert version that supports deconvolution, you will also see Deconvolution (Stars) and Deconvolution (Objects), however if these models are not available locally and you don't have a version of GraXpert that is capable of downloading them they will not be shown.

Next follows an area for operation-specific parameters. If you select Denoising you will find a widget to control strength, if you select Background Extraction you will find widgets for smoothing and the type of background extraction to perform, and so on. These are described in more detail below.

The Advanced setting provides a means of controlling the inferencing batch size and whether or not to use GPU acceleration. It is generally a good idea to leave the batch size at the default : it sets how many chunks of the image can be processed in parallel, and increasing it may result in out-of-memory errors.

Astuce

The GraXpert models use ONNX to perform inferencing. This is a hardware- agnostic Machine Learning framework that provides various runtimes targeting different types of acceleration hardware. There is a specific NVidia runtime, a DirectML runtime that provides generic Windows GPU acceleration, a MacOS runtime and Linux runtimes for NVidia, AMD and Intel GPUs. The script will automatically try to install the most suitable runtime for your hardware, however in the event that it does not work there is always a CPU runtime to revert to - simply disable the GPU acceleration checkbox and please submit a bug report as this is new functionality and we are especially keen to hear of any issues relating to the ONNX runtime selection.

Finally, the UI provides *Apply* and *Close* buttons that perform the expected functions and a *Model Manager* button that opens a tool to discover and download versions of the AI models.

When opening the GUI, it will detect whether a single image or a sequence is loaded. If a single image is loaded it will process it and update the result in Siril, whereas if a sequence is loaded it will process all the selected frames of the sequence and create a new sequence with a prefix dependent on the operation being performed.

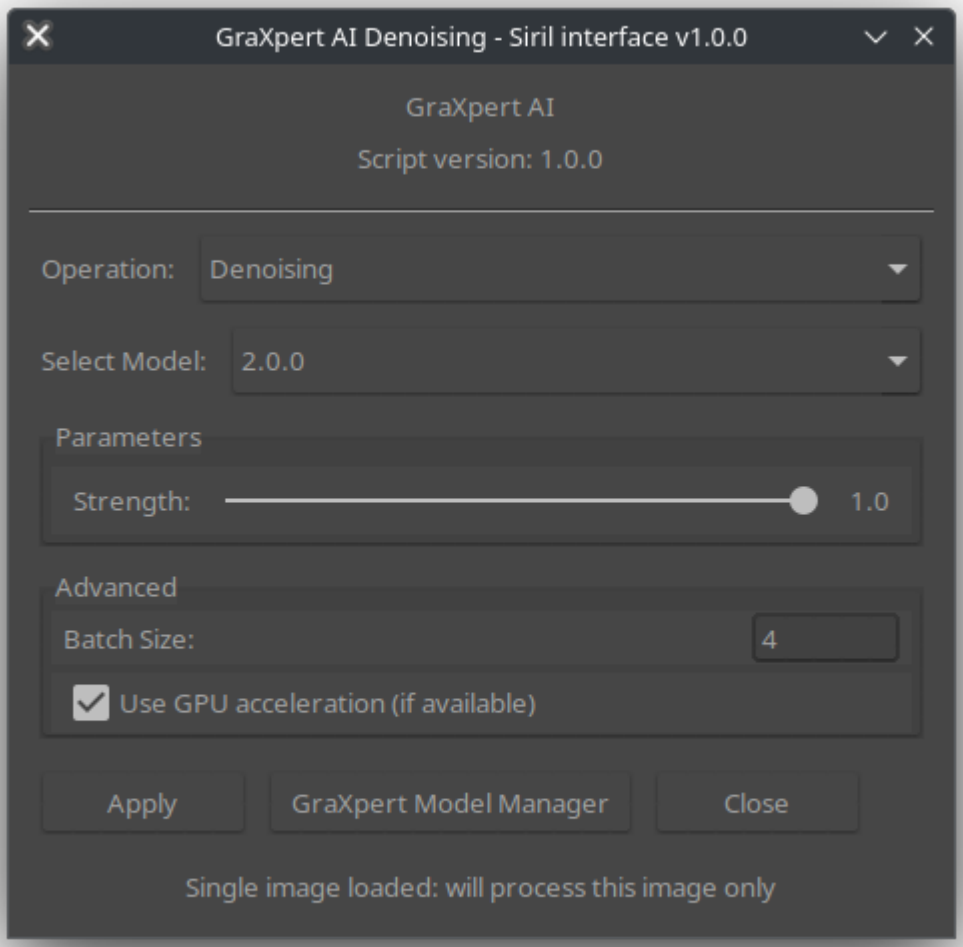


FIG. 78 – GraXpert script interface, showing denoising options.

Avertissement

In Siril 1.4, Python scripting cannot subscribe to Siril event notifications so it is not possible for the script to detect if the image has been closed or if a change between a single image and a sequence occurs after opening the tool. Therefore if you do this, you should close and reopen the script GUI.

10.8.3 Extraction de gradient

The GraXpert background extraction AI model offers two parameters :

— Smoothing - this determines the smoothness of the background model that is generated.

— Correction type - this determines the type of correction that is applied : you can choose from subtraction or division.

Note that the GraXpert script does not provide an interface to the classical background removal algorithms provided by GraXpert. RBF is already natively available in Siril, and Spline and Kriging do not generally provide better performance than RBF.

10.8.4 Déconvolution

From version 3.1 onwards, GraXpert supports deconvolution. With the models downloaded, Siril supports both modes provided by GraXpert :

— Déconvolution d'objets

— Déconvolution stellaire

These modes can be selected using the *Operation* dropdown.

Both deconvolution modes have two parameters, *Strength*, which takes values between 0.0 (no deconvolution) and 1.0 (maximum deconvolution) and *Image FWHM (in pixels)*, which also takes values between 0.0 (prioritises fine structure retention) and 10.0 (prioritise deconvolution ability). If you set it to zero you can see a very (too) small effect on processed image, if so set it higher.

10.8.5 Débruitage

To use the GraXpert AI denoising model, select *Denoising* in the Operations dropdown. This operation has only one parameter, *Strength*, which takes values between 0.0 (no denoising) and 1.0 (maximum denoising).

Denoising can take a little time, especially for large images and / or if using a CPU for inferencing. However once the denoising is done the denoised image is cached, so experimenting with different strengths is extremly quick.

10.8.6 Gestionnaire de modèles

The script provides a model manager to discover and download model versions. Whereas inferencing is done directly in the script, model download requires use of the actual GraXpert program because the models are stored in an Amazon S3 instance and GraXpert stores its credentials internally. The model manager will show radio buttons for each model that the installed version of GraXpert is capable of downloading. For v3.0.x this means denoising and background extraction; for v3.1.0-rcx this also adds stellar and object deconvolution.

The list of remotely available models is shown. If you want to discover models for a different AI operation, change the radio button to the operation you want a model for and click *Refresh*. A model can be downloaded by selecting it in the list and clicking *Download selected model*. Note that the download may take some time in the case of large models or slow network connections and GraXpert provides little feedback, so be patient. The models are stored in the GraXpert user data directory.

Avertissement

Do not move the GraXpert models from the GraXpert data directory structure where they are saved by the model manager. This is where the script expects to find them and it will ignore models put in any other location.

Once a model is downloaded it will automatically be populated in the model version dropdown in the main UI when the relevant AI operation is selected.

10.8.7 Scripts

The GraXpert-AI.py script provides a command-line interface for use in scripting. This uses the *pyscript* command as shown below.

Ligne de commande Siril

```
pyscript scriptname.py [script_argv]
```

Exécute un script Siril en python

Le nom du script doit être fourni en tant que premier argument. S'il n'est pas trouvé dans le répertoire de travail actuel, les chemins d'accès aux scripts définis par l'utilisateur spécifiés dans les préférences et le répertoire local siril-scripts seront recherchés. Tous les arguments suivants seront traités comme des arguments de script et transmis au script sous forme de vecteur d'arguments. Notez que le script spécifique doit prendre en charge la lecture des entrées à partir du vecteur d'arguments

Paramètres d'opération

Les paramètres suivants permettent de définir l'opération à effectuer :
- **-denoise** spécifie le débruitage
- **-deconv_obj** spécifie la déconvolution d'objet
- **-deconv_stellar** spécifie la déconvolution d'étoile
- **-bge** spécifie l'extraction du fond de ciel

Paramètres Générales

The following parameters are recognised in all modes :
- **-h / --help** prints help to the Siril console and exits
- **-model** specifies the model version to use (e.g. -model="3.0.2") If this argument is omitted it defaults to the latest available version.
- **-batch** specifies the AI batch size
- **-gpu** spécifie l'utilisation du GPU (ceci est l'option par défaut)
- **-nogpu** spécifie d'utiliser uniquement l'interface CPU
- **-listmodels** causes the script to list the available model versions for the specified operation and exit

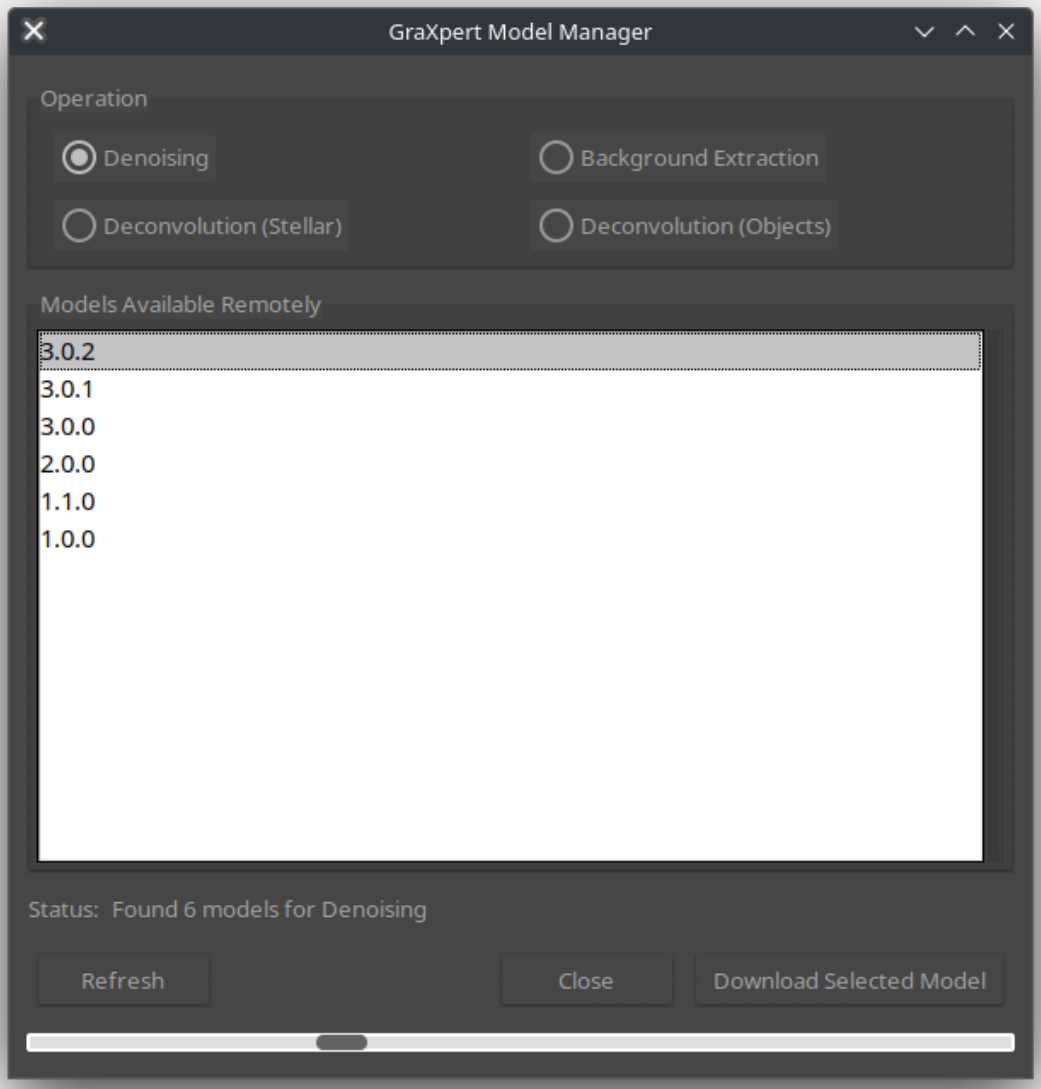


FIG. 79 – Interface de script GraXpert, affichant le gestionnaire de modèles.

Astuce

Note that this script does not provide a -seq parameter as it expects to work on a sequence already loaded in Siril. When scripting, this can be achieved using the *load_seq* command first to load the desired sequence.

Paramètres de l'extraction du fond de ciel

- **-smoothing=** This option sets the smoothing parameter, between 0.0-1.0
- **-correction=** This option sets the correction type and may be either "subtraction" or "division"
- **-keep-bg** This option specifies that the modelled background should be saved as well. For a filename such as myfile.fit, the background will be saved as myfile_bg.fit

Paramètre de débruitage

- **-strength=** Cette option définit l'intensité du paramètre, entre 0,0 et 1,0

Paramètre de déconvolution

The same parameters are available for both stellar and object deconvolution :
- **-strength** This option sets the deconvolution strength, between 0.0-1.0
- **-psfsize** This option sets the PSF size, as described above

Exemples

pyscript GraXpert_AI.py -denoise -model='3.0.1'

Ou, si vous l'appelez depuis un autre script Python :

siril.cmd("pyscript", "GraXpert-AI.py", "-denoise", "-model='3.0.1'")

10.9 Extraction des canaux

10.9.1 Séparation des canaux

This function creates three monochrome images from a 3-channel color image, depending on the configured color space. For RGB, it's simply splitting the file in three. For the others, it involves computation of the equivalent color space, either HSL (hue-saturation-lightness), HSV (hue-saturation-value) or CIELAB.

Astuce

Si aucun nom n'est donné à un canal, celui-ci n'est pas extrait.

Ligne de commande Siril

```
split file1 file2 file3 [-hsl | -hsv | -lab]
```

Divise l'image couleur chargée en trois fichiers distincts (un pour chaque couleur) et les enregistre dans les fichiers **file1**.fit, **file2**.fit et **file3**.fit. Un dernier argument peut être fourni, **-hsl**, **-hsv** ou **lab** pour effectuer une extraction TSL, TSV ou CieLAB. Si aucune option n'est fournie, l'extraction est de type RVB, ce qui signifie qu'aucune conversion n'est effectuée

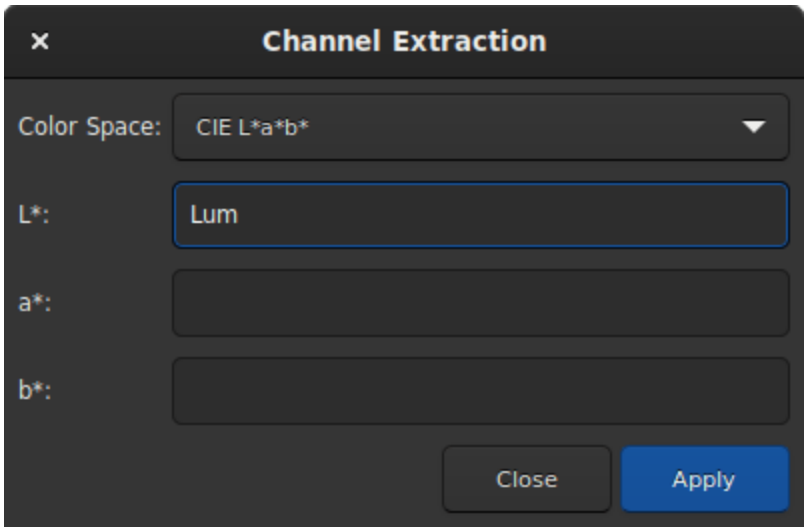


Fig. 80 – Boite de dialogue Séparer les canaux.

10.9.2 Séparer les canaux CFA

CFA signifie color filter array (matrice de filtres couleur). Ce terme est souvent utilisé pour décrire le contenu d'une image couleur à un canal, chaque pixel correspondant à des valeurs acquises derrière un filtre sur le capteur. Il s'oppose aux images dématricées.

L'ouverture d'une image CFA dans Siril est nécessaire pour le prétraitement, comme l'élimination du signal dark avant l'interpolation de l'image en couleur à trois canaux. Nous pouvons également utiliser les informations du filtre de couleur pour extraire des images comme celle-ci :

- **Séparer les canaux CFA** : quartes images sont créées à partir de l'image CFA, chacune représentant un filtre de la matrice de Bayer, donc en général R.fit, G1.fit, G2.fit et B.fit. Ceci est utile si le but est de traiter séparément les différentes couleurs de l'image.
- **Extraire Ha** : l'utilisation d'un filtre H-alpha avec une image de caméra couleur (OSC : on-sensor color, ou one-shot color camera) signifie que seuls les pixels avec des filtres rouges seront utiles, donc en général seulement un quart d'entre eux. Cette fonction crée une nouvelle image qui ne contient que les pixels associés au filtre rouge documenté dans la matrice de Bayer de l'image.
- **Extraire Ha/OIII** : pour les caméras OSC, des filtres qui laissent passer les photons des longueurs d'onde H-alpha et O-III sont apparus. Cette extraction crée deux images : une image à partir des pixels rouges comme l'extrait Ha, et une image combinant les pixels verts et bleus en une seule pour O-III. Les deux images correspondent à la moitié de la définition de l'image d'entrée.

> **Note**
>
> Une question est fréquemment posée sur la raison pour laquelle les images Ha et OIII sont de tailles différentes et sur la manière dont elles sont réparties. Cette note tente d'apporter une réponse à cette question. Dans un capteur d'images couleur, les pixels sont recouverts d'une matrice de filtres très fine appelée matrice de filtres couleur (CFA) ou matrice de Bayer. La disposition des pixels filtrés correspond à un certain nombre de modèles : RVVB, VBVR, etc.

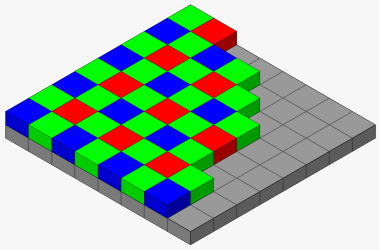


FIG. 81 – Image originale par Cburnett, sous la licence CC BY-SA 3.0.

Parmi ces pixels, seuls les pixels R sont sensibles à Ha. Nous commençons donc par séparer tous les pixels rouges dans une image Ha. Étant donné que seul 1 élément CFA sur 4 est rouge, les dimensions de l'image Ha sont inférieures de moitié à celles de l'image originale.
Les pixels restants, V et B, sont tous sensibles à l'OIII. La sensibilité des pixels filtrés V à l'OIII est différente de la sensibilité des pixels filtrés B à l'OIII, mais ils représentent la même scène et sont répartis uniformément, de sorte que l'intensité moyenne doit être la même.

$$V_i = V_{io} \times \frac{3 \times \overline{V_o}}{2 \times \overline{V_o} + \overline{B_o}}$$

$$B_i = B_{io} \times \frac{3 \times \overline{B_o}}{2 \times \overline{V_o} + \overline{B_o}}$$

Où B_i est le $i^{ème}$ pixel bleu, B_{io} est le $i^{ème}$ pixel bleu d'origine et $\overline{B_o}$ est la moyenne de tous les pixels bleus d'origine (et de même pour les pixels verts).
Jusqu'à présent, nous disposons d'un ensemble égalisé de pixels V et B avec des lacunes où les pixels R ont été supprimés. Enfin, nous utilisons l'interpolation bilinéaire pour estimer les valeurs des pixels R et obtenir une image OIII de taille normale.

Note

L'option de rééchantillonnage Ha/OIII permet de gérer la sortie de l'extraction Ha/OIII. Pas de rééchantillonnage produit une image OIII en pleine résolution et une image Ha en demi résolution ; sur-échantillonner Ha rééchantillonne l'image Ha d'un facteur 2 pour correspondre à l'image OIII ; sous-échantillonner OIII rééchantillonne l'image OIII d'un facteur 2 pour correspondre à l'image Ha.
Il se peut que vous souhaitiez utiliser le drizzle pour mettre à l'échelle les données Ha au lieu de les mettre à l'échelle. Comme le drizzle est une méthode d'empilement, dans ce cas vous devez utiliser *seqextract_HaOiii* pour extraire le Ha et le OIII de chaque image de la séquence, et ensuite empiler les images OIII de la manière habituelle et les images Ha avec un drizzle x2.

— **Extraire Vert** : pour la photométrie, il est souvent utile de ne traiter que la partie verte de l'image CFA, parce qu'elle est plus sensible et qu'elle a deux pixels à moyenner, ce qui réduit encore le bruit. Bien entendu, l'image créée voit également sa définition divisée par deux.

Note

Ces fonctions ne fonctionnent que si la matrice de Bayer a été correctement documentée par le logiciel d'acquisition et si le format d'image le supporte, donc en général FITS ou SER.

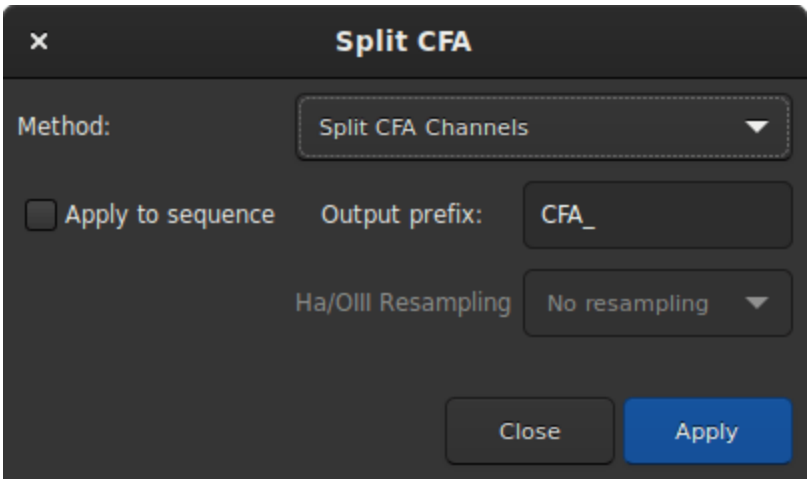


Fig. 82 – Boite de dialogue Séparer les canaux CFA.

Avertissement

Cela ne fonctionne pas avec d'autres matrices de filtres que les matrices de Bayer, comme le Fujifilm X-TRANS.

10.9.3 Couches des ondelettes

Cet outil extrait les différents plans de l'image en appliquant le processus d'ondelettes. Chaque plan est enregistré dans une image et l'ensemble des images peut être lu comme une séquence. Vous pouvez choisir jusqu'à 9 couches pour le calcul des ondelettes et le type d'algorithme est soit Linéaire, soit BSpline. Ce dernier est généralement préféré.

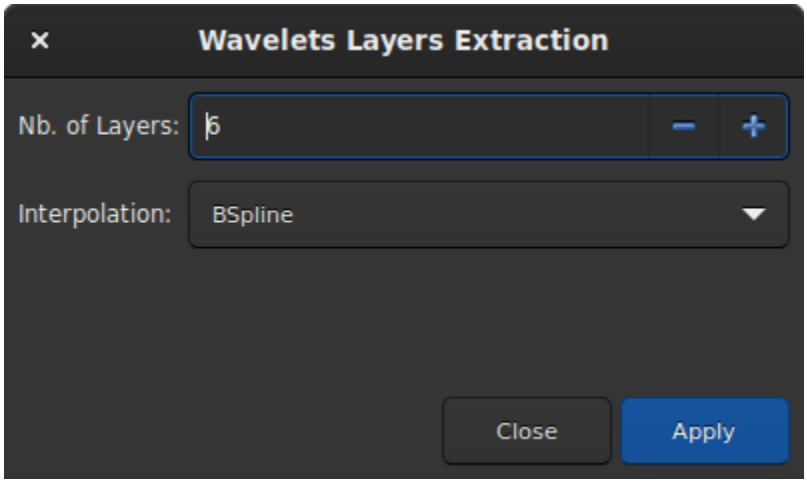


Fig. 83 – Boite de dialogue Extraction des Couches des Ondelettes.

La décomposition se fait à travers un certain nombre de couches de détail définies à des échelles caractéristiques croissantes et une couche résiduelle finale, qui contient les structures restantes non résolues.

FIG. 84 – Image originale de M45 (avec la permission de V. Cohas).

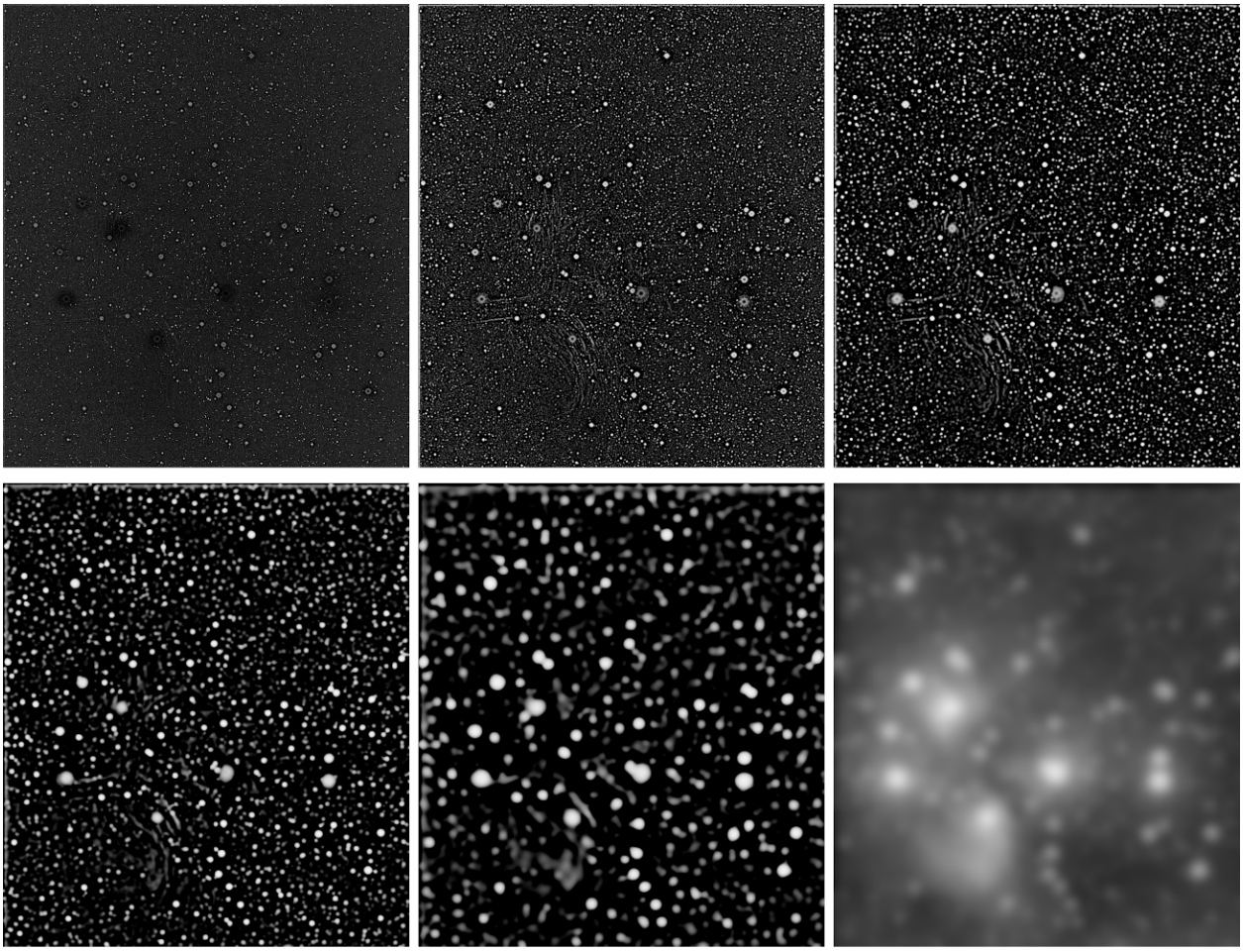


FIG. 85 – 6 plans extraits.

10.10 Linear Match

L'appariement linéaire consiste à trouver une fonction linéaire qui fait correspondre au mieux (au sens de la méthode des moindres carrés) l'intensité des pixels d'une image à celle d'une image de référence. Il s'agit d'un moyen simple et rapide d'équilibrer les histogrammes de différentes images.

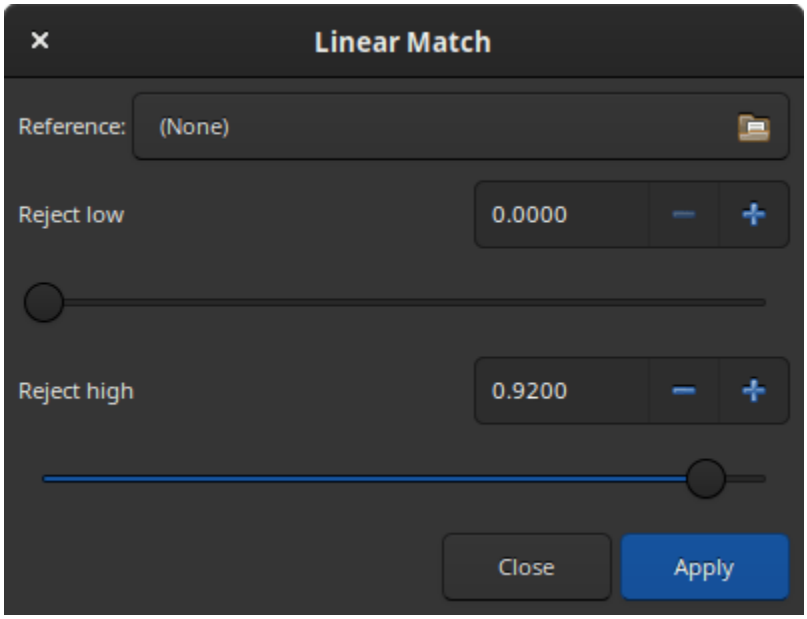


FIG. 86 – Boite de dialogue Linear match

Le bouton *Réference* vous permet de choisir l'image de référence.

Les curseurs *Rejet bas* et *Rejet haut* permettent d'exclure les valeurs des pixels situés dans les queues gauche et droite des distributions d'intensités. Ils sont définis comme des quantiles, dans l'intervalle [0, 1]. Par exemple, la valeur par défaut pour haut est 0.92, ce qui signifie que les 8 % de pixels les plus lumineux seront exclus de l'ajustement pour trouver les coefficients de correspondance linéaire.

Avertissement

L'image et la référence doivent être alignées avant d'appliquer une correspondance linéaire. Dans le cas contraire, il n'y a aucune raison de supposer que les intensités de leurs pixels sont corrélées.

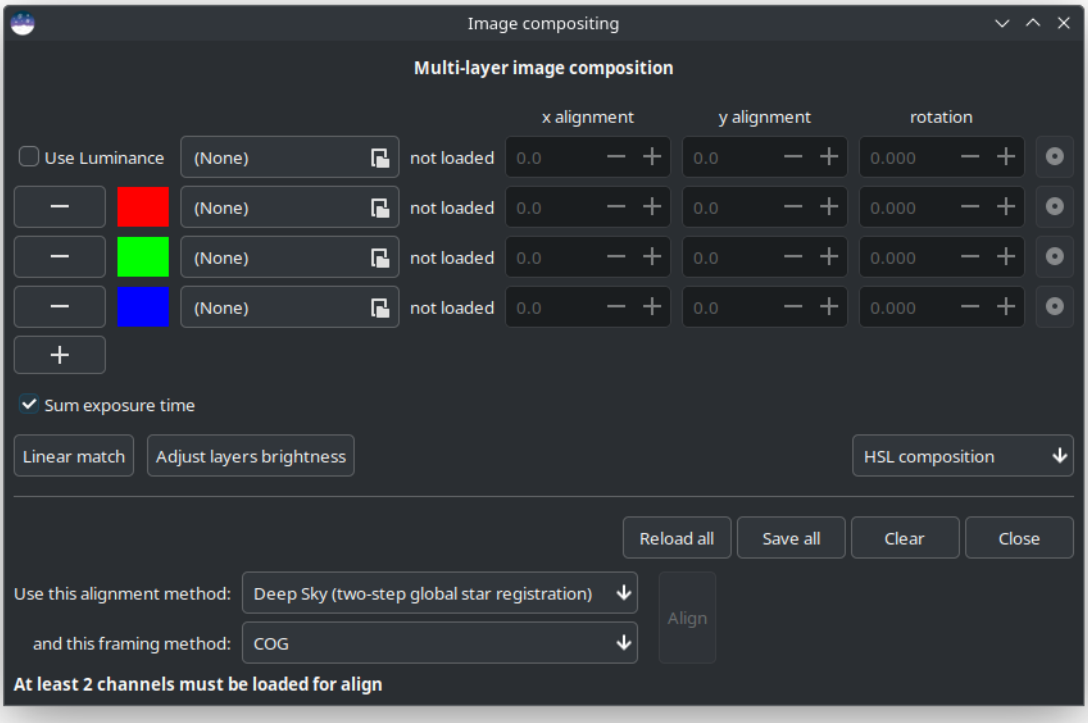
Ligne de commande Siril

```
linear_match reference low high
```

Calcule et applique une fonction linéaire entre une image de **référence** et l'image chargée.

L'algorithme ignore tous les pixels de référence dont les valeurs se situent en dehors de la plage [**low**, **high**]

10.11 Composition RVB



L'outil de composition RVB vous permet d'assembler jusqu'à 8 images monochromes pour former une seule image couleur. L'outil a été mis à jour pour donner accès à toutes les meilleures méthodes d'alignement disponibles dans Siril, ainsi qu'à une méthode d'alignement manuel unique à cet outil.

Le fonctionnement de cet outil est assez simple, il suffit de charger les images et de leur attribuer une couleur. Le premier champ, optionnel, est réservé à la couche de luminance. Une fois la couche de luminance chargée, vous pouvez l'intégrer ou non dans la composition grâce au bouton *Utiliser la Luminance*. Chaque couleur peut être personnalisée en cliquant dessus et en choisissant une nouvelle. Lorsque plus de 3 images (ou 4 s'il y a de la luminance) sont chargées, ou lorsque les couleurs assignées sont des mélanges de couleurs plutôt que des R, V et B purs, il peut être nécessaire d'ajuster la luminosité de chaque canal. Le bouton *Ajuster la luminosité des canaux* effectue cette opération automatiquement.

Lors de la composition de différents canaux, en particulier à partir de filtres à bande étroite, les histogrammes peuvent être fortement désynchronisés. Une correction initiale de base peut être effectuée en appliquant une correspondance

linéaire à l'aide du bouton *Linear match*. Cela fonctionne de la même manière que la même entrée dans le menu *Traitement de l'image*, mais est adapté à cet outil particulier. La couche supérieure chargée (habituellement Luminance ou Rouge) est prise comme canal de référence, et tous les autres canaux sont automatiquement mis en correspondance linéaire avec elle. Ceci est particulièrement important si l'on assigne des couleurs non pures à certains canaux, car chaque canal sera potentiellement mélangé à chaque canal RVB dans le résultat, et il ne sera plus possible d'ajuster séparément les histogrammes des canaux d'entrée en ajustant les histogrammes R, V et B de la composition.

Il est possible d'activer ou non l'option*Additionner les temps d'exposition*. Cette option permet à l'utilisateur d'additionner ou non les temps d'exposition des images individuelles, en stockant le résultat dans l'en-tête FITS.

Note

Pour le binning et les dimensions de l'image, la première image chargée détermine la taille de l'image de sortie. Si vous avez des images de tailles différentes, vous devez toujours charger la plus grande en premier. Si vos images sont différentes en raison du binning, mais avec le même champ de vision, l'outil de composition augmentera la taille des plus petites images lorsqu'elles seront chargées pour qu'elles correspondent à la taille de la première image chargée. Cela est utile pour les images L-RVB communes prises avec les filtres de couleur en bin2. Cela signifie également que si deux images n'ont pas été prises avec le même capteur, il est peu probable qu'elles aient le même champ de vision et le même échantillonnage de pixels après le rééchantillonnage de l'image. La capacité de cet outil à aligner de telles images dépendra de la différence d'échelle et de la méthode d'alignement utilisée.

Trois espaces colorimétriques sont disponibles pour le rendu de la composition :
- TSL (pour teinte, saturation, luminosité)
- TSV (pour teinte saturation valeur)
- CIE L*a*b*

et sont laissés au choix de l'utilisateur.

Once the composition is finished, it is possible to do a linear match of the channels using the *Linear match* button.

Si les images ne sont pas alignées entre elles, il est possible de les aligner. Les algorithmes suivants sont disponible :
- **Ciel profond (alignement global des étoiles en deux étapes)** : aucune sélection n'est requise pour cette méthode. Cette méthode génère une homographie très précise à 8 degrés de liberté entre les images.
- **Planétaire (alignement par motifs)** : vous devez dessiner une sélection autour de l'objet que vous souhaitez aligner. Un contraste suffisamment élevé est nécessaire pour que l'algorithme fonctionne correctement. Cette méthode n'aligne que les décalages d'image par translation, elle ne peut pas compenser la rotation. Elle convient pour les images planétaires, mais pour les images du ciel profond, il est recommandé d'utiliser la méthode d'alignement global des étoiles.
- **KOMBAT** : comme pour l'alignement par motif, vous devez dessiner une sélection autour de l'objet que vous souhaitez aligner. Cette méthode n'aligne également que les décalages d'images par translation. Elle est beaucoup plus rapide que l'autre méthode, mais peut s'avérer moins efficace lorsque les motifs d'image des différents canaux sont très différents (par exemple, deux canaux planétaires filtrés avec des filtres IR et UV peuvent ne pas présenter suffisamment de similitudes de motifs pour que KOMBAT puisse les faire correspondre).
- **Alignement manuel** : cette méthode est propre à l'outil de composition RVB. Lorsqu'ils sont sélectionnés, les boutons de rotation changent de fonction : ils affichent des informations sur les décalages d'alignement et vous permettent d'entrer les décalages souhaités en x et y ainsi que la rotation. La rotation est centrée par défaut sur le centre de l'image, mais un centre de rotation personnalisé peut être choisi pour chaque calque à l'aide du bouton de centrage situé à droite de chaque ensemble de commandes de calque. Cette méthode peut s'avérer utile si vous avez des images qui ne se prêtent pas à l'alignement par l'une des autres méthodes, ou si vous souhaitez affiner les résultats.

Vous pouvez choisir la méthode de gestion du cadre qui sera appliquée à vos calques alignés. Les méthodes suivantes sont possibles :
- **Premier canal** - cette méthode permet de cadrer toutes les autres images sur le premier canal (généralement la luminance ou le rouge). Cette méthode est particulièrement utile pour les compositions LRGB.

— **MIN** - cette méthode applique le cadrage minimum : le plus grand rectangle comprenant uniquement les régions couvertes par toutes les images d'entrée sera cadré.
— **COG** - cette méthode sélectionne un cadrage optimal pour la composition sur la base du centre de gravité des pixels alignés de l'image.

Note

Les méthodes de cadrage ne s'appliquent pas à l'alignement par motifs. Cette méthode est toujours basée sur le premier canal. Si cette méthode d'alignement est sélectionnée, la méthode de cadrage choisie sera ignorée.

Astuce

Lors de l'alignement, l'outil de composition RVB aligne les images chargées en mémoire. Cela signifie que si vous décidez que vous n'êtes pas satisfait de l'alignement et que vous voulez essayer une autre méthode ou un autre cadrage, vous devez recharger toutes les images pour revenir à leur état d'origine avant l'alignement. Le bouton *Tout recharger* est prévu à cet effet.

Si vous préférez composer vos images d'une autre manière, par exemple en utilisant Pixel Math, vous pouvez toujours utiliser l'outil de composition RVB comme moyen rapide d'aligner les images. Après l'alignement, presser *Enregistrer tout* sauvegardera toutes les images alignées avec leurs noms de fichiers originaux préfixés par *comp_*. Ces images peuvent ensuite être utilisées dans l'outil de votre choix et, contrairement à ce qui se passe si vous créez une séquence avec vos empilements et que vous l'alignez de la manière habituelle, le nom du fichier indique clairement quelle est l'image en question.

Cet outil est par nature graphique et n'est pas exactement reproduit dans une commande Siril, mais il existe des commandes qui peuvent être utilisées pour réaliser une partie du processus :
— *Alignement* peut aligner une séquence contenant les images empilées,
— The *rgbcomp* command can compose a RGB image from 3 separate R, G and B images or 4 L, R, G, B images, however the images must be pre-aligned and it cannot assign mixed colors to the input images.
— *Pixelmath* peut également être utilisé dans des scripts pour combiner les canaux.
— Il existe également un script de composition RVB intégré qui permet d'automatiser en partie le processus.

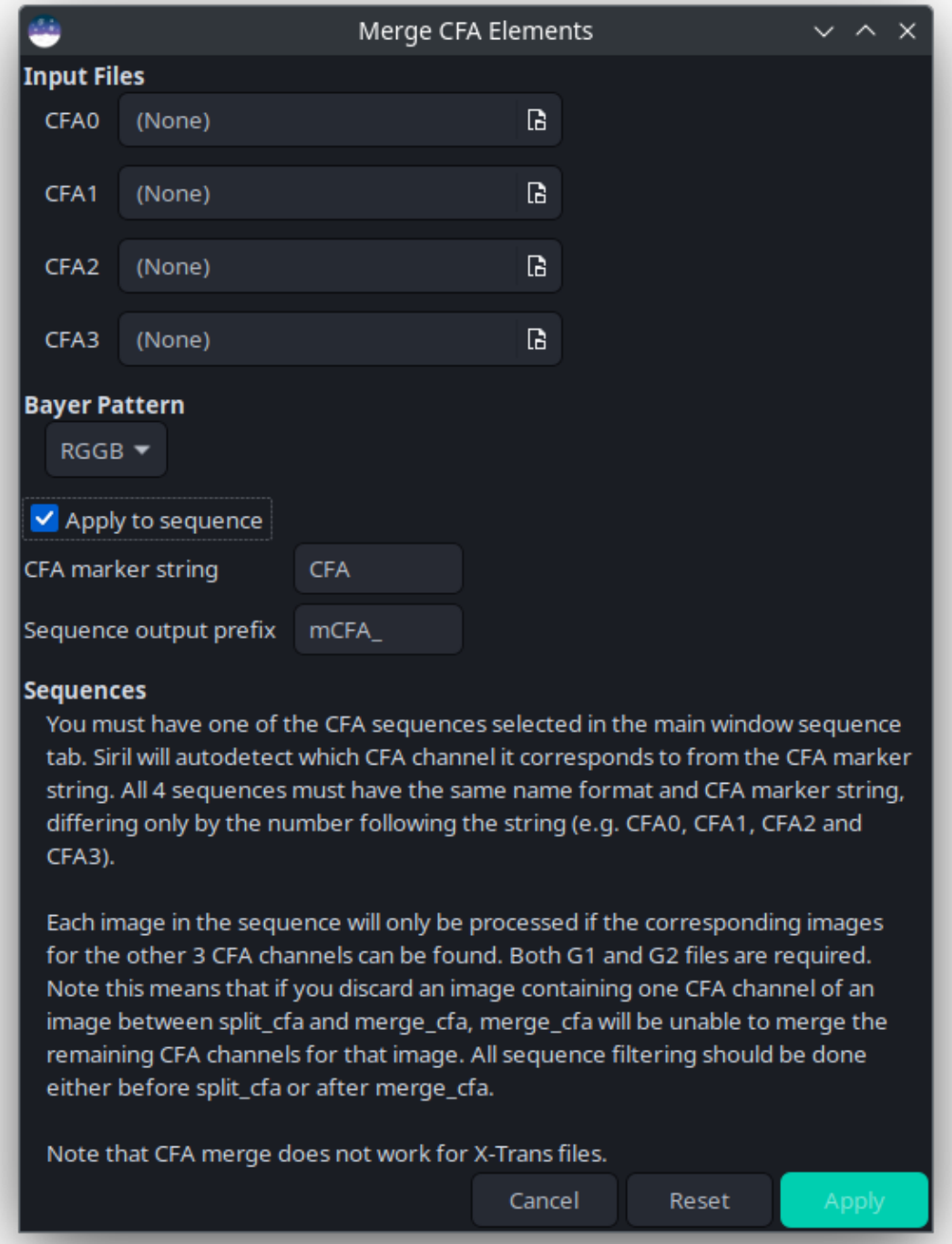
Ligne de commande Siril

```
rgbcomp red green blue [-out=result_filename] [-nosum]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename] [-nosum]
```

Crée une composition RVB à l'aide de trois images indépendantes, ou une composition LRVB à l'aide de l'image de luminance optionnelle et de trois images monochromes ou d'une image couleur. L'image résultante est appelée composed_rgb.fit ou composed_lrgb.fit à moins qu'un autre nom ne soit fourni dans l'argument optionnel. Un autre argument optionnel, **-nosum**, indique à Siril de ne pas additionner les temps d'exposition. Cela a un impact sur les mots-clés FITS tels que LIVETIME et STACKCNT

10.12 Fusionner les canaux CFA

Le but de cet outil est de combiner plusieurs images monochromes qui ont été préalablement extraites d'un capteur CFA (avec le menu *Extraction → Séparer les canaux CFA...* par exemple). L'outil fusionne les images séparées des canaux rouge, vert (x2) et bleu en une seule image composite appelée image CFA.


Merge CFA Elements
Input Files
CFA0
(None)
CFA1
(None)
CFA2
(None)
CFA3
(None)
Bayer Pattern
RGGB
Apply to sequence
CFA marker string
CFA
Sequence output prefix
mCFA_
Sequences
You must have one of the CFA sequences selected in the main window sequence tab. Siril will autodetect which CFA channel it corresponds to from the CFA marker string. All 4 sequences must have the same name format and CFA marker string, differing only by the number following the string (e.g. CFA0, CFA1, CFA2 and CFA3).
Each image in the sequence will only be processed if the corresponding images for the other 3 CFA channels can be found. Both G1 and G2 files are required. Note this means that if you discard an image containing one CFA channel of an image between split_cfa and merge_cfa, merge_cfa will be unable to merge the remaining CFA channels for that image. All sequence filtering should be done either before split_cfa or after merge_cfa.
Note that CFA merge does not work for X-Trans files.
Cancel
Reset
Apply

Avertissement

Cet outil est dédié aux images provenant d'une matrice de Bayer et ne peut donc pas fonctionner avec des images provenant de fichiers issue des capteurs X-Trans des appareils photo Fuji.

Le dialogue est divisé en trois parties différentes :

- **Input files** : Select the images containing the CFA0, CFA1, CFA2 and CFA3 Bayer subpatterns. If these have been produced using Siril's "Split CFA" function they will have the CFA prefix.
- **Motif de Bayer** : Définit l'en-tête de motif de Bayer à appliquer au résultat. Il doit correspondre au motif de Bayer de l'image à partir de laquelle les sous-canaux de Bayer originaux ont été extraits.
- La partie séquence, en bas, permet de traiter des séquences entières en reconstituant une séquence d'images CFA. Un clic sur le bouton *Appliquer à la séquence* affiche un texte d'aide pour procéder correctement. Ce texte est reporté dans l'infobulle suivante. Deux options sont disponibles :
  - **Marqueur d'entrée de séquence** : Préfixe d'identification utilisé pour indiquer le numéro de canal CFA. Il doit correspondre au préfixe de séquence utilisé lors de l'exécution du processus de séparation des canaux CFA (par défaut : `CFA_`).
  - **Préfixe de sortie de la séquence** : Préfixe des noms d'images résultant du processus de fusion des canaux CFA. Par défaut, c'est `mCFA_`.

Astuce

You may have any of the CFA0, CFA1, CFA2 or CFA3 sequences selected in the main window sequence tab : Siril will determine which one it is from the prefix and the number.
Your separate sub-CFA sequences should have been processed in exactly the same way; they **must** have the same sequence name format and CFA marker string, differing only but the number 0-3 following the CFA marker string (e.g. CFA0_bg_pp_lights, CFA1_bg_pp_lights, CFA2_bg_pp_lights and CFA3_bg_pp_lights).
Each image in the sequence will only be processed if the corresponding images for the other 3 CFA channels can be found. Both Green subchannel images 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.

Ligne de commande Siril

```
merge_cfa file_CFA0 file_CFA1 file_CFA2 file_CFA3 bayerpattern
```

Builds a Bayer masked color 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'

Ligne de commande Siril

```
seqmerge_cfa sequencename0 sequencename1 sequencename2 sequencename3 bayerpattern [-
↪prefixout=]
```

Fusionne 4 séquence d'image pour reconstruire la matrice de Bayer. Les séquence sont spécifié dans les arguments **sequencename0**, **sequencename1**, **sequencename2** et **sequencename3**.

The Bayer pattern to be reconstructed must be provided as the second argument as one of RGGB, BGGR, GBRG or GRBG (the order of the Bayer channels must match the order of the specified sequences).

Note : les 4 séquences **doivent** être présente et avoir les mêmes dimension, profondeur de bit et nombre d'images.

Le nom de la séquence de sortie commence par le préfixe "mCFA_" sauf indication contraire avec l'option **-prefixout=**

10.13 Pixel Math

L'un des outils les plus puissants de Siril est Pixel Math. Il vous permet de manipuler les pixels des images à l'aide de fonctions mathématiques. De la simple addition ou soustraction à des fonctions plus avancées, comme la MTF, Pixel Math est un outil parfait pour le traitement des images astronomiques.

Cette page a pour but de décrire entièrement l'outil, pour voir des exemples détaillés, veuillez vous référer à l'excellent tutoriel sur le site.

La fenêtre est divisée en 5 parties.

1. La première, comprenant 3 zones de texte recevant les formules mathématiques. Seule la première est utilisée si vous souhaitez produire une image monochrome. Décochez le bouton *Utiliser une seule expression RVB/K* pour produire une sortie RVB.
2. La seconde est la zone des variables avec la sélection des *Functions* et des *Operators*. Chaque variable est une image qui doit être chargée au préalable avec le bouton +. Vous pouvez cliquer sur la fonction et/ou l'opérateur souhaité pour le faire apparaître dans la saisie de la formule.
3. Le troisième, le champ **paramètres**, permet à l'utilisateur de définir des paramètres séparés par `,`. Par exemple, si vous définissez des paramètres avec l'expression `factor =0.8, K=0.2`, toutes les occurrences de `factor` et `K` dans la formule ci-dessus seront remplacées par 0.8 et 0.2 respectivement. La formule `Ha * factor + OIII * K` sera donc évaluée à `Ha * 0.8 + OIII * 0.2`.
4. Le champ **output** est réservé à la mise à l'échelle de l'image dans une plage donnée et à l'activation ou non de l'option*Additionner les temps d'exposition*. Cette dernière option permet à l'utilisateur d'additionner ou non les temps d'exposition des images individuelles, en stockant le résultat dans l'en-tête FITS. Il est nécessaire de développer le cadre avant de l'utiliser.
   Partie mise à l'échelle de la boite de dialogue Pixel Math
5. Enfin, la zone **préréglages** permet à l'utilisateur de réutiliser des formules précédemment sauvegardées à l'aide du bouton situé à droite des zones de formule. Il est nécessaire de développer le cadre avant de l'utiliser. Double-cliquez sur la formule pour la copier dans la bonne entrée.

10.13.1 Utilisation

Nom des variables

Par défaut, il est possible de charger 10 images simultanément. Chaque image reçoit un nom de variable commençant par I suivi d'un nombre de 1 à 10. Cependant, si l'image chargée contient le mot-clé `FILTER`, alors la valeur de ce dernier devient le nom de la variable par défaut. Bien entendu, il est toujours possible de le modifier en double-cliquant dessus.

`$T` is a reference to the current image, meaning the image on which the PixelMath operations will be applied.

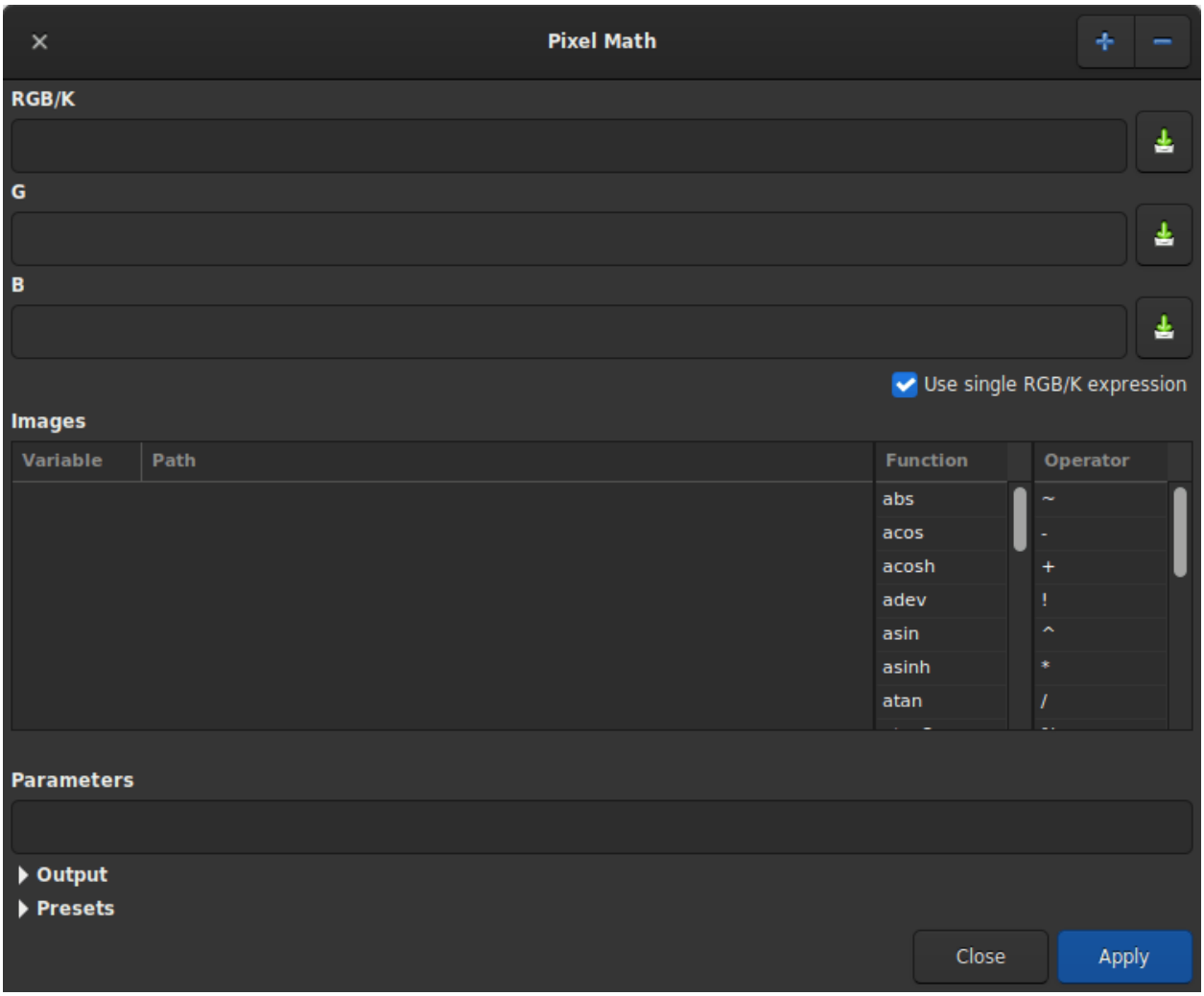


FIG. 87 – La boîte de dialogue Pixel Math est illustrée à l'ouverture

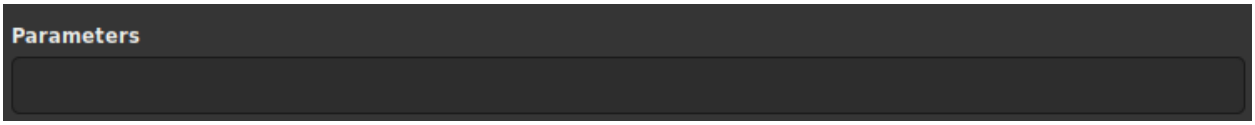


FIG. 88 – Partie paramètres de la boite de dialogue Pixel Math

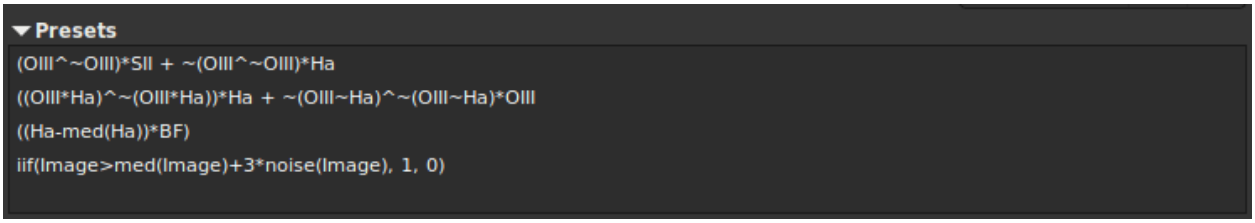


FIG. 89 – Préréglages Pixel Math

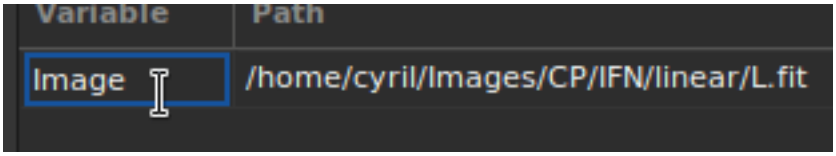


FIG. 90 – Il est possible de changer le nom de la variable.

Astuce

It is possible to use the currently loaded image by using the `$T` token. Note, however, that unlike other programs, the expression `$T[i]`, with `i=0,1,2`, is not recognized.

Exemples

Prenons une image monochrome de galaxies. Il s'agit d'une image linéaire vue avec le mode d'affichage Auto ajustement.

L'expression suivante :

```
iif(Image>med(Image)+3*noise(Image), 1, 0)
```

produira un masque d'étoiles.

Ligne de commande Siril

```
pm "expression" [-rescale [low] [high]] [-nosum]
```

Cette commande évalue l'expression donnée en argument comme dans l'outil PixelMath. L'expression complète doit être entre guillemets et les variables (qui sont des noms d'images, sans extension, situés dans le répertoire de travail dans ce cas) doivent être entourées du symbole $, par exemple "$image1$ * 0.5 + $image2$ * 0.5". Un maximum de 10 images peut être utilisé dans l'expression.

L'image peut être redimensionnée avec l'option **-rescale** suivie des valeurs **low** et **high** dans l'intervalle [0, 1]. Si aucune valeur basse ou haute n'est fournie, les valeurs par défaut sont 0 et 1. Un autre argument optionnel, **-nosum**, indique à Siril de ne pas additionner les temps d'exposition. Cela a un impact sur les mots-clés FITS tels que LIVETIME et STACKCNT

10.13.2 Fonctions

Il y a deux type de fonctions. Celles qui s'appliquent directement sur les pixels et celles qui s'appliquent sur l'image entière (comme les fonctions statistiques).

FIG. 91 – Image d'origine.

FIG. 92 – Après la formule ci-dessus.

Tableau 1: Fonction orientée sur les pixels

Fonction	Cas d'utilisation	Définition
abs	abs (x)	Valeur absolue de x.
acos	acos (x)	Arc cosinus de x.
acosh	acosh (x)	Arc cosinus hyperbolique de x.
asin	asin (x)	Arc sinus de x.
asinh	asinh (x)	Arc sinus hyperbolique de x.
atan	atan (x)	Arc tangente de x.
atan2	atan2 (y, x)	Arc tangente de y/x.
atanh	atanh (x)	Arc tangente hyperbolique de x.
ceil	ceil (x)	Arrondi x à l'entier supérieur.
cos	cos (x)	Cosinus de x.
cosh	cosh (x)	Cosinus hyperbolique de x.
e	e	La constante e=2.718282...
exp	exp (x)	Fonction exponentielle.
fac	fac(x)	Fonction factorielle.
iif	iif(cond, expr_true, expr_false)	Fonction conditionnelle (ou fonction if inline). Retourne *expr_true* si *cond* a une valeur non nulle. Retourne *expr_false* si *cond* est égale à zéro.
floor	floor (x)	Plus grand entier inférieur ou égal à x.
ln	ln (x)	Logarithme népérien de x.
log	log (x)	Logarithme base 10 de x.
log10	log10 (x)	Logarithme base 10 de x.
log2	log2 (x)	Logarithme en base 2 de x.
max	max (x,y)	Fonction maximum.
min	min (x,y)	Fonction minimum.
mtf	mtf (m, x)	Fonction de transfert des demi-tons (MTF) de x pour un paramètre d'équilibre des demi-tons m dans la plage [0, 1].
ncr	ncr (x, y)	Fonction de combinaisons.
npr	npr (x, y)	Fonction permutation.
pi	pi	La constante =3.141592...
pow	pow (x, y)	Fonction exponentielle.
sign	sign (x)	Sign de x : $+1$ si $x > 0$ 1 si $x < 0$ 0 si $x = 0$.
sin	sin (x)	Sinus de x.
sinh	sinh (x)	Sinus hyperbolique de x.
sqrt	sqrt (x)	Racine carrée de x.
tan	tan (x)	Tangente de x.
tanh	tanh (x)	Tangente hyperbolique de x.
trunc	trunc (x)	Partie entière tronquée de x.

Tableau 2 – Fonctions statistiques

Fonction	Cas d'utilisation	Définition
adev	adev (Image)	Déviation moyenne absolue de l'image.
bwmv	bwmv (Image)	Moyenne pondérée de l'image.
height	height (Image)	Hauteur en pixels de l'image spécifiée.
mad	mad (Image)	Écart absolu médian de l'image. L'utilisation de mdev est également possible.
max	max (Image)	Maximum d'un pixel de l'image.
moyenne	mean (Image)	Moyenne de l'image.
med	med (Image)	Médiane de l'image. L'utilisation de median est également possible.
min	min (Image)	Minimum d'un pixel de l'image.
noise	noise (Image)	Estimation du bruit Gaussien dans l'image.
sdev	sdev (Image)	Déviation standard de l'image.
width	width (Image)	Largeur en pixel d'une image spécifiée.

10.13.3 Opérateurs

Tableau 3 – Opérateurs

Opérateur	Cas d'utilisation	Définition
~	~x	Opérateur d'inversion des pixels.
-	-x	Opérateur unaire Moins (changement de signe).
+	+x	Opérateur unaire Plus.
!	!x	Opérateur logique NOT.
^	x ^ y	Opérateur d'exponentielle.
*	x * y	Opérateur de multiplication.
/	x / y	Opérateur de division.
%	x % y	Opérateur modulo.
+	x + y	Opérateur d'addition.
-	x - y	Opérateur de soustraction.
<	x < y	Opérateur relationnel Plus Petit.
<=	x <= y	Opérateur relationnel inférieur ou égal.
>	x > y	Opérateur relationnel Plus Grand.
>=	x >= y	Opérateur relationnel plus grand que ou égal.
==	x == y	Opérateur relationnel Égal À.
!=	x != y	Opérateur relationnel N'est Pas Égal À.
&&	x && y	Opérateur logique AND.
||	x || y	Opérateur logique OR.

CHAPITRE 11

Astrométrie

L'astrométrie est la science qui étudie de la position et le mouvement des objets célestes. L'astrométrie est essentielle dans l'astrophotographie moderne avec des logiciels de captures tels que N.I.N.A, Ekos, APT ou autres, qui réalisent une résolution des images dans le but d'obtenir une solution astrométrique, c'est-à-dire connaître précisément la position de l'image sur le ciel. L'astrométrie peut aussi être utilisé durant le traitement, comme outil de calibration photométrique par exemple.

11.1 Astrométrie

La résolution astrométrique est une étape majeure dans le traitement des images astronomiques. Elle permet d'associer les images à des coordonnées célestes, ce qui permet de savoir quel objet se trouve dans le champ de vision observé. De nombreux outils de Siril, tels que les outils d'étalonnage des couleurs spectrophotométriques ou photométriques (SPCC ou PCC), ont besoin de connaître les coordonnées de l'image avec une précision suffisante pour fonctionner.

L'astrométrie dans Siril peut être réalisée de deux différentes manières :

— En utilisant l'outil dédié accessible par le menu *Outils* → *Astrométrie* → *Résolution astrométrique*, ou en utilisant le raccourci `Ctrl` + `Shift` + `A`.

— Utilisation de la commande `résolution astrométrique`, introduite dans Siril 1.2.

Ligne de commande Siril

```
platesolve [-force] [image_center_coords] [-focal=] [-pixelsize=]
platesolve ... [-noflip] [-downscale] [-order=] [-radius=] [-disto=]
platesolve ... [-limitmag=[+-]] [-catalog=] [-nocrop]
platesolve ... [-localasnet [-blindpos] [-blindres]]
```

Résolution astrométrique de l'image chargée.

Si l'image a déjà été résolue, rien ne sera fait, à moins que l'argument **-force** ne soit passé pour forcer une nouvelle résolution. Si les mots clés WCS ou d'autres métadonnées de l'image sont erronés ou manquants, des arguments doivent être passés :

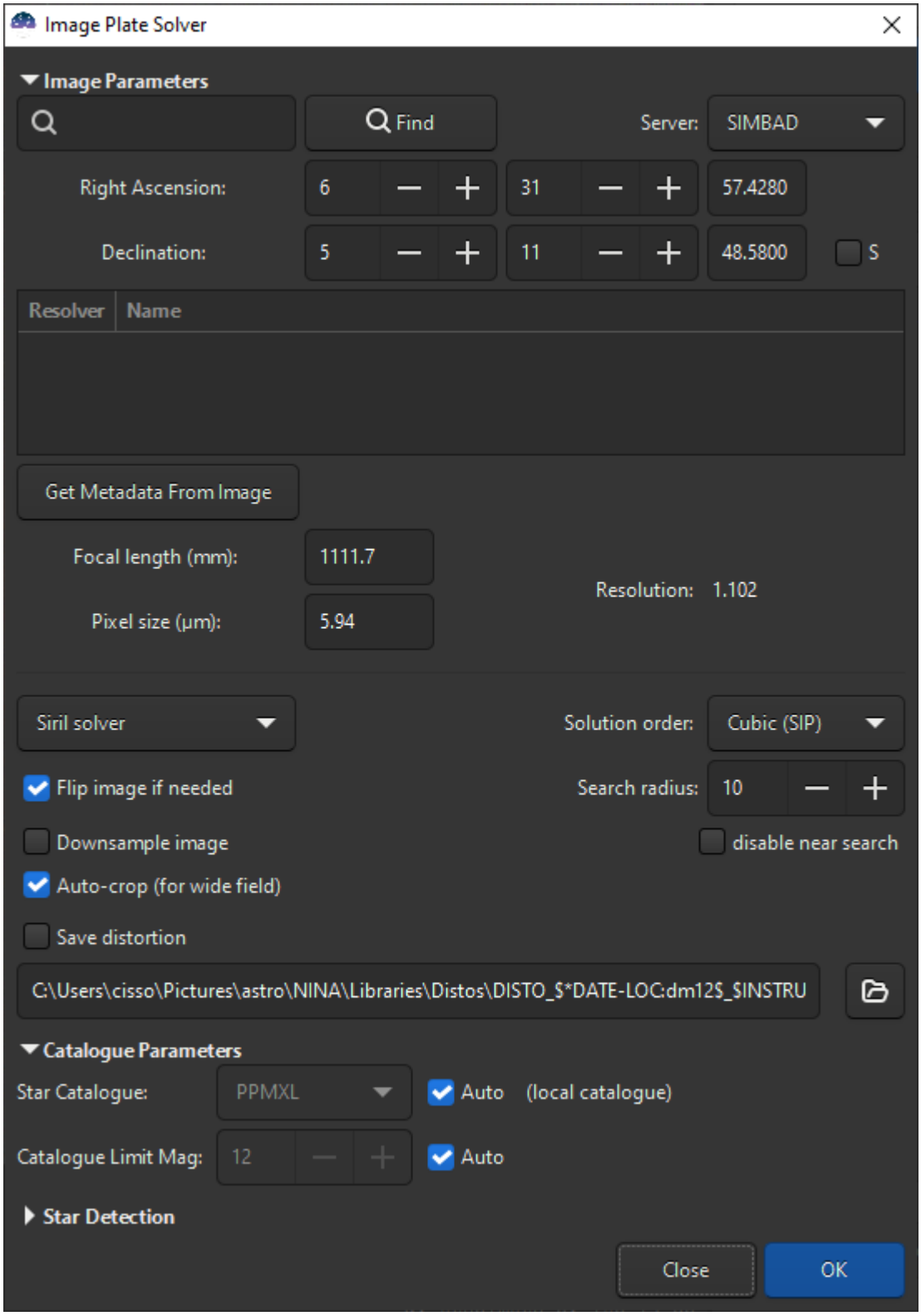


FIG. 1 – Dialogue résolution astrométrique

les coordonnées approximatives du centre de l'image peuvent être fournies en degrés décimaux ou en degrés/heure minute seconde (J2000 avec des séparateurs de deux points), avec les valeurs d'ascension droite et de déclinaison séparées par une virgule ou un espace (non obligatoire pour astrometry.net).

La longueur focale et la taille des pixels peuvent être passées avec **-focal=** (en mm) et **-pixelsize=** (en microns), en remplaçant les valeurs de l'image et des paramètres. Voir aussi les options pour résoudre en aveugle avec Astrometry.net en local

Sauf si **-noflip** est spécifié, si l'image est détectée comme étant à l'envers, elle sera retournée.

Pour une détection plus rapide des étoiles dans les grandes images, il est possible de sous-échantillonner l'image avec **-downscale**.

La résolution peut prendre en compte les distorsions en utilisant la convention SIP avec des polynômes jusqu'à l'ordre 5. La valeur par défaut est prise dans les préférences astrométriques. Elle peut être modifiée avec l'option **-order=** qui donne une valeur entre 1 et 5.

Lors de l'utilisation des catalogues locaux du solveur Siril ou avec Astrometry.net local, si la résolution initiale n'est pas réussie, le solveur cherchera une solution dans un cône de rayon spécifié avec l'option **-radius=**. Si aucune valeur n'est fournie, le rayon de recherche est pris dans les préférences d'astrométrie. La recherche Siril par proximité peut être désactivée en passant une valeur de 0 (ne peut pas être désactivée pour Astrometry.net).

Vous pouvez sauvegarder la solution courante comme fichier de distorsion avec l'option **-disto=**.

Les images peuvent être résolues par Siril à l'aide d'un catalogue d'étoiles et de l'algorithme d'alignement global ou par la commande locale solve-field d'astrometry.net (activée avec **-localasnet**).

Options du solveur astrométrique de Siril :

La magnitude limite des étoiles utilisées pour la résolution astrométrique et le PCC est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement avec une valeur absolue positive pour la magnitude limite.

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 catalog given in argument, with possible values : tycho2, nomad, localgaia, gaia, ppmxl, brightstars, apass.

Si le champ de vision calculé est supérieur à 5 degrés, la détection des étoiles sera limitée à une zone recadrée autour du centre de l'image, à moins que l'option **-nocrop** ne soit passée.

Options du solveur Astrometry.net :

Les options **-blindpos** et/ou **-blindres** permettent de résoudre en aveugle la position et la résolution respectivement. Vous pouvez utiliser ces options pour résoudre une image dont la position et l'échantillonnage sont totalement inconnus

Depuis la version 1.2, la résolution astrométrique peut être effectuée par deux algorithmes différents. Le premier était le seul dans Siril jusqu'à cette version, il est basé sur l'algorithme d'alignement global des étoiles, essayant d'aligner les images sur une image virtuelle d'un catalogue avec le même champ de vision. Le second est nouveau, il utilise un programme externe appelé `solve-field` de la suite Astrometry.net, installé localement. Pour les plateformes Windows, la façon la plus simple de l'obtenir est d'utiliser ansvr.

Since version 1.3, Siril internal solver will also look within a cone around the initial target coordinates if it did not find a match. This is only available when using *local star catalogues*. The log will show this information when that happens :

```
Initial solve failed
Attempting a near solve with a radius of 10.0 degrees
```

Les solutions astrométriques nécessitent la prise en compte de quelques paramètres, comme l'échantillonnage de l'image. La fenêtre de l'outil permet de rassembler ces paramètres, nous allons maintenant voir comment les remplir correctement.

11.1.1 Paramètres de l'image

Coordonnées de la cible

Il est plus facile et plus rapide de trouver une solution astrométrique lorsque l'on sait à peu près où l'on cherche. Le solveur de Siril, qui compare un catalogue avec l'image, a besoin de connaître approximativement la position du centre de l'image pour obtenir l'extrait du catalogue. Astrometry.net possède tous les catalogues dont il a besoin localement, il peut donc les parcourir tous pour trouver une solution, mais il est bien sûr beaucoup plus rapide de lui dire par où commencer.

Les logiciels d'acquisition contrôlent souvent aussi le télescope de nos jours et devraient connaître les coordonnées approximatives de l'endroit où l'image a été prise. Dans ce cas, en utilisant un format FITS, ces coordonnées seront fournies dans les métadonnées de l'image, l'en-tête FITS. Ce n'est pas toujours le cas, et il est clair que ce n'est pas le cas lorsque des images DSLR RAW sont créées au lieu de FITS.

When opening the plate solver window, 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).

Le champ texte en haut à gauche de la fenêtre est le champ de recherche, son but étant de convertir un nom d'objet en ses coordonnées. En appuyant sur `ENTRER` ou en cliquant sur le bouton *Chercher*, l'objet sera d'abord recherché dans les catalogues d'annotations locaux. S'il n'est pas trouvé, une requête Web sera effectuée pour obtenir ses coordonnées. Plusieurs résultats peuvent être trouvés pour le nom saisi, ils seront affichés dans la liste ci-dessous. La sélection d'un résultat met à jour les coordonnées en haut de la liste, le premier résultat étant sélectionné par défaut.

Il est également possible de choisir le serveur sur lequel vous voulez exécuter la requête, cela ne change pas beaucoup les résultats, mais parfois l'un d'entre eux peut être en ligne, alors les autres agiraient comme une sauvegarde, entre CDS, VizieR et SIMBAD (par défaut).

> **Note**
>
> Si l'objet n'est pas trouvé, veuillez essayer avec le nom complet ou avec le nom d'un catalogue. Les catalogues d'annotations contiennent quelques noms communs, les services en ligne aussi, mais pas tous, et ils ne trouvent pas de réponses partielles. Par exemple, pour la Nébuleuse de la Bulle, veuillez entrer `NGC 7635` ou `bubble nebula` en anglais, et non `bubble`.

Les champs de coordonnées sont remplis automatiquement, mais il est possible de définir les vôtres. N'oubliez pas de cocher la case *S* si l'objet que vous recherchez est situé dans l'hémisphère sud du ciel (déclinaisons négatives).

Echantillonnage de l'images

L'échantillonnage de l'image est le paramètre le plus important pour la résolution astrométrique. Donné en secondes d'arc par pixel dans notre cas, il représente le degré de zoom sur le ciel de l'image, donc la largeur du champ à rechercher.

Elle est dérivée de deux paramètres : la longueur focale et la taille des pixels. Ils sont souvent disponibles dans les métadonnées de l'image. Lorsqu'elles ne sont pas disponibles dans l'image, les valeurs stockées dans les paramètres sont utilisées. Les valeurs des images et des préférences peuvent être définies à l'aide de la boîte de dialogue *Information*. Dans tous les cas, vérifiez la valeur affichée avant la résolution et corrigez-la si nécessaire. Si une solution astrométrique est trouvée, la longueur focale et la taille de pixel par défaut seront écrasées. Ce comportement peut être désactivé dans les paramètres.

Avertissement

Si le binning a été utilisé, il doit être spécifié dans l'en-tête FITS, mais cela peut prendre deux formes : la taille du pixel peut rester la même et le multiplicateur de binning doit être utilisé pour calculer l'échantillonnage, ou la taille du pixel est déjà multipliée par le logiciel d'acquisition. Selon le cas, l'une ou l'autre de ces formes peut être choisie dans les préférences ou dans la fenêtre *Information*.

La taille des pixels est indiquée dans les spécifications des caméras astronomiques, et peut généralement être trouvée sur le Web pour les appareil photo numérique ou autres. Le nombre de capteurs est limité et la plupart d'entre eux sont connus.

La focale dépend de l'instrument principal, mais aussi du backfocus et des lentilles correctrices ou de zoom utilisées. Donnez une valeur aussi proche que possible de ce que vous pensez être la focale effective, si une solution astrométrique est trouvée, la longueur focale calculée sera donnée dans les résultats et vous pourrez la réutiliser dans votre logiciel d'acquisition et pour les utilisations futures de l'outil.

Lorsque l'un des champs est mis à jour, l'échantillonnage est recalculé et affiché dans la fenêtre (appelée ici "résolution"). Veillez à ce que la valeur soit aussi proche que possible de la réalité.

Astuce

Les données écrites en orange dans l'interface graphique indiquent des valeurs qui n'ont pas pu être extraites de l'en-tête de l'image. Cela ne signifie pas qu'elles sont erronées (elles auraient pu être chargées à partir des valeurs de préférences et être valides) ou obligatoires (vous pourriez résoudre le problème à l'aveugle), cette couleur est juste là pour différencier les valeurs lues à partir de l'en-tête.

11.1.2 Paramètres du solver

Depuis Siril 1.2, Siril peut utiliser deux solveurs différents, son solveur interne et*une installation locale d'Astrometry.net*. L'interface diffère selon que l'un ou l'autre est sélectionné dans la liste déroulante dédiée.

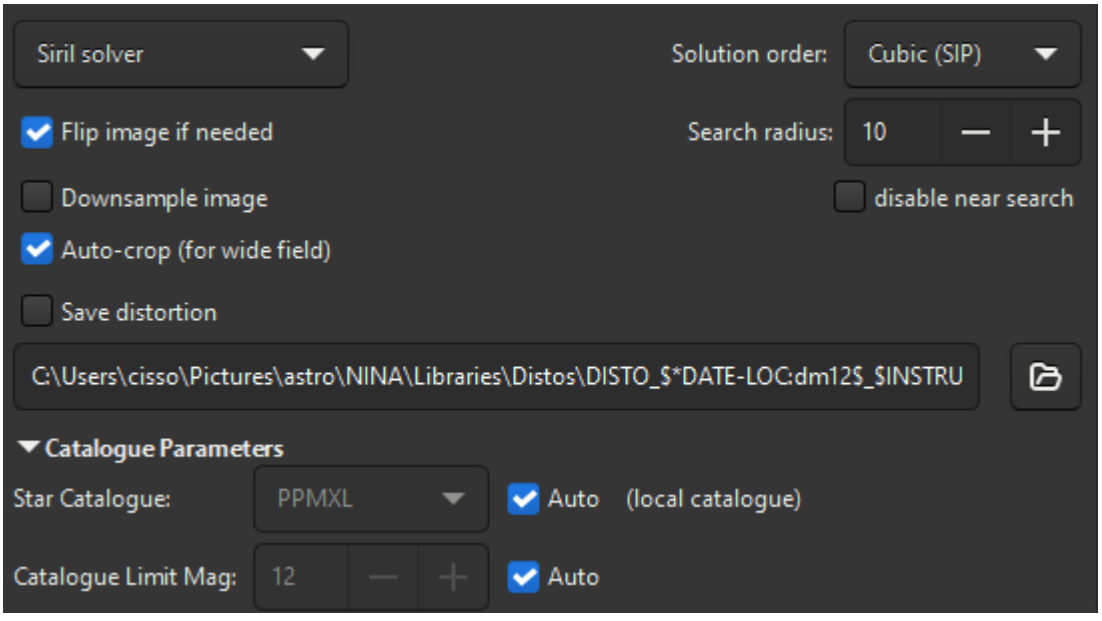


Fig. 2 – Options du solver interne de Siril

Distorsions

L'option*Ordre*, sélectionnée via la liste déroulante, spécifie l'ordre de la solution de résolution astrométrique. En sélectionnant `Linéaire`, le solveur essaiera d'ajuster une solution en supposant qu'il n'y ait pas de distorsions dans l'image (c'est-à-dire que le champ est optiquement plat). Cependant, cette hypothèse peut ne pas être vraie en présence d'aberrations optiques (mauvais backfocus, pas d'aplanisseur de champ, etc...). Depuis la version 1.3, le solver peut essayer d'ajuster des distorsions polynomiales jusqu'au cinquième ordre, en suivant la convention SIP. Par défaut, il utilise les distorsions `Cubiques (SIP)`, ce qui devrait convenir à la plupart des cas d'utilisation. Cette valeur par défaut peut être modifiée dans les*Preferences*. Cette option est disponible avec les deux solveurs.

You can also choose to save distortions to a wcs file by ticking the *Save distortion* box. Use the filechooser below to specify its path. If you have defined a master distortion path in the *preferences*, its path will be shown by default. This file can then be used to undistort images during *Global Registration* and *2pass Registration*

Rayon de la recherche

When using *local star catalogues* are installed or when using *Astrometry.net*, a cone around the target position will be searched. The size of this cone in degrees can be changed using the *Search radius* control, which default value can be changed in the *Preferences*. For Siril solver, this feature can be disabled by ticking the *disable near search* box.

Solver Astrometry.net à l'aveugle

Lorsque le solveur Astrometry.net est sélectionné, deux options supplémentaires sont disponibles :

1. *Ignorer la position (à l'aveugle)* permet d'ignorer la position spécifiée dans le champ des coordonnées de la cible.
2. *Ignorer la résolution (à l'aveugle)* permet d'ignorer la résolution calculée à partir de la taille du pixel et de la longueur focale.

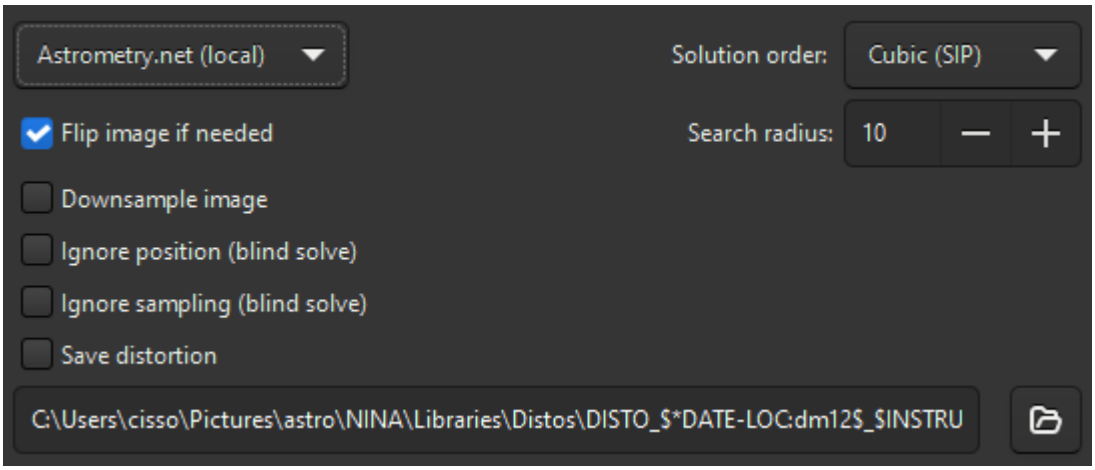


Fig. 3 – Options du solveur Astrometry.net

Utilisez ces deux options conjointement si la position et la résolution de l'image sont totalement inconnues.

Avertissement

Il n'y a pas de magie ici. Pour résoudre n'importe quel champ de vision, vous devez avoir installé les*indexes* nécessaires qui couvrent le champ de vision réel de l'image à résoudre.

Autres paramètres

Enfin, il y a trois boutons à bascule en bas du cadre :

1. L'option *Sous-échantillonner l'image* réduit l'image d'entrée pour accélérer la détection des étoiles. L'inconvénient est qu'elle peut ne pas trouver assez d'étoiles ou donner une solution astrométrique moins précise. La taille de l'image de sortie reste inchangée.
2. Si l'image est détectée comme étant à l'envers par la solution astrométrique, avec l'option *Retourner l'image si nécessaire* activée, elle sera retournée à la fin. Cela peut être utile selon le logiciel de capture, si l'image n'a pas la bonne orientation lorsqu'elle est affichée dans Siril (voir plus *d'explications*).
3. Lorsque l'option *Recadrage automatique (pour grand champ)* est appliquée, elle n'effectue une coupe que dans le centre de l'image. Ceci n'est fait que pour les images à grand champ (plus de 5 degrés) où les distorsions loin du centre sont suffisamment importantes pour tromper l'outil. Ignoré pour les résolutions d'astrometry.net.

11.1.3 Paramètre du catalogue

This section is relevant for Siril internal solver only. Several online catalogues can be used and also two catalogues that can be installed locally for faster and more reliable operation.

Utilisation des catalogues en ligne

Par défaut, cette section est grisée car tout est réglé sur automatique. En décochant la case automatique, il est toutefois possible de choisir le catalogue en ligne utilisé pour l'astrométrie, qui peut dépendre de la résolution de l'image. Le choix se fait entre :

— TYCHO2, un catalogue contenant les positions, les mouvements propres et les données photométriques en deux couleurs pour 2 539 913 des étoiles les plus brillantes de la Voie lactée.
— NOMAD, une simple fusion des données des catalogues Hipparcos, Tycho-2, UCAC2, Yellow-Blue 6, et USNO-B pour l'astrométrie et la photométrie, complété par le catalogue 2MASS en proche infrarouge. Ce jeu de données de près de 100 Go contient des données astrométriques et photométriques pour environ 1,1 milliard d'étoiles.
— 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. This represents the state of the art in accurate astrometry data and provides spectral data for a wide range of sources which is used in SPCC.
— PPMXL, un catalogue de positions, de mouvements propres, de photométrie 2MASS- et optique de 900 millions d'étoiles et de galaxies.
— Bright Stars, un catalogue d'étoiles qui répertorie toutes les étoiles d'une magnitude stellaire de 6,5 ou moins, soit à peu près toutes les étoiles visibles à l'œil nu depuis la Terre. Le catalogue contient 9 110 objets.
— APASS, un catalogue d'étoiles qui répertorie toutes les étoiles de magnitude stellaire 6,5 ou plus, c'est-à-dire à peu près toutes les étoiles visibles à l'œil nu depuis la Terre. Le catalogue contient 9 110 objets.

> **Note**
>
> Une connexion internet est requise pour utiliser ces catalogues en ligne.

Tous ces catalogues sont disponibles via l'outil d'accès au catalogue VizieR, CDS, Strasbourg, France (DOI :10.26093/cds/vizier). La description originale du service VizieR a été publiée en 2000, A&AS 143, 23.

L'option *Magnitude limite* permet de limiter la magnitude des étoiles récupérées dans le catalogue. La valeur automatique est calculée à partir de la résolution de l'image.

Using the local KStars catalogues (NOMAD)

Avec la version 1.1, à partir de juin 2022, il était possible de s'appuyer sur un catalogue d'étoiles installé localement, pour un fonctionnement déconnecté ou plus résilient. Le catalogue d'étoiles que nous avons trouvé le plus adapté à nos besoins est celui de KStars. Il est en fait composé de quatre catalogues (documentés ici dans KStars), deux d'entre eux n'étant pas directement distribués dans les fichiers d'installation de base de KStars :

- **namedstars.dat**, les étoiles les plus brillantes, elles ont toutes un nom
- **unnamedstars.dat**, aussi des étoiles brillantes, mais jusqu'à magnitude 8
- **deepstars.dat**, des étoiles moins brillantes extraites du catalogue Tycho-2 de 2,5 millions d'étoiles les plus brillantes, jusqu'à la magnitude 12,5
- **USNO-NOMAD-1e8.dat** un extrait de l'énorme catalogue NOMAD limité aux informations photométriques B-V et au mouvement propre des étoiles sous forme de binaire compact, jusqu'à la magnitude 18.

This catalogue can be used for plate solving and for PCC.

En comparant ces catalogues avec NOMAD en ligne, on peut facilement constater que de nombreuses étoiles sont manquantes. Si le nombre d'étoiles trouvées est insuffisant pour votre champ étroit, vous pouvez toujours utiliser les requêtes à distance. Une bonne chose à vérifier lorsque les catalogues sont installés est de mettre en évidence les étoiles de l'image qui seront utilisées pour le PCC, celles qui sont disponibles avec des informations photométriques dans les catalogues, en utilisant la commande *conesearch*.

```
conesearch -phot
```

Using the local Gaia DR3 Catalogues

In version 1.4 support for a local extract of the Gaia DR3 catalogue has been added. This *astrometric* Gaia extract differs from the other catalogues in that it is focused on astrometry as applied to plate solving, so instead of defining a limiting magnitude the catalogue has been designed to provide an even coverage of stars in each level 8 HEALPixel (you can read more about HEALPixels here). This avoids including excessive numbers of stars in densely populated parts of the sky and ensures that sufficient fainter stars are included in emptier HEALPixels to support accurate plate solving.

The default catalogue will be selected at startup based on two criteria :

- If a local catalogue is available it is preferred to the remote equivalent
- If a Gaia catalogue (local or remote) is available it is preferred to NOMAD

> **Astuce**
>
> Although the offline Gaia DR3 extract can be used with the *conesearch* command (by specifying the catalogue name *localgaia*) it is not really designed for this purpose. In busy regions of the sky the limiting magnitude will be relatively bright and therefore many stars visible in your images will not be contained in the catalogue whatever limiting magnitude you set. For annotation cone searches in such regions you will be better served by specifying the catalogue name *gaia* instead, which will conduct a remote cone search using the Vizier mirror of the Gaia DR3 archive. This is both fast (for a remote query) and comprehensive.

This catalogue can be used only for plate solving. A *photometric* Gaia extract also exists, and can be used for the SPCC, it is documented on the *SPCC page*.

Télécharger KStars NOMAD

Les deux premiers fichiers sont disponibles dans KStars source, le catalogue Tycho-2 dans un paquet Debian et le catalogue NOMAD dans les fichiers KStars également, comme documenté dans ce petit article pour l'installation de KStars. Il dispose de plusieurs miroirs dans le monde entier comme indiqué dans les 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. So those *.xz* files are compressed, make sure you uncompress them with a suitable software before copying them.

Direct download links are available here (right click on each file name on the left and save the links).

In case the link is unavailable and to make it faster from anywhere, it is also distributed with bittorrent, using this torrent file or the following magnet link.

Télécharger les données d'astrométrie Gaia DR3

The Gaia DR3 local astrometric extract is available from Zenodo It comes with a sha256 checksum for the compressed archive.

Full details, plus a citation reference and a checksum for the uncompressed archive for extra paranoia, can be found at the Zenodo record. The specification of the Gaia DR3 catalogue extracts and their file format is documented here (PDF).

Astuce

For convenience, an offline version of the Gaia DR3 astrometric catalogue can be installed using the Catalogue Installer script in the *Scripts* menu. This catalogue takes up 1.5GB once installed compared with 2.1GB for the KStars NOMAD catalogue and provides higher precision astrometry and even density of stars in all regions, for more reliable and accurate plate solving. **It is strongly recommended to install and use this catalogue for all astrometry purposes within Siril**.

Installation dans Siril

Les fichiers peuvent être placés n'importe où et leurs chemins donnés à Siril dans les paramètres, mais il y a un emplacement par défaut pour les quatre fichiers : `~/.local/share/kstars/` sous Linux. Ils peuvent être liés à cet emplacement pour éviter les copies inutiles. Les paramètres peuvent maintenant être modifiés à partir de la ligne de commande, en utilisant la commande *set*.

Lorsqu'il est disponible et lisible, Siril n'utilisera pas le service Web pour récupérer des données astrométriques ou photométriques. Consultez les messages dans l'onglet log ou sur la console pour vérifier que les fichiers catalogues sont utilisés comme prévu.

À partir de Siril 1.4.0, Siril va d'abord chercher dans les *catalogues d'annotations locales* pour trouver les coordonnées d'un objet passé dans la fenêtre de la résolution astrométrique, afin de localiser le centre de l'image. Cela signifie que, si vous avez installé les catalogues d'étoiles locaux, vous pouvez résoudre vos images sans connexion Internet. Bien sûr, cela ne devrait être nécessaire que si le logiciel d'acquisition n'a pas enregistré les coordonnées de la cible dans l'en-tête FITS, ou si vous utilisez le format de fichier SER qui ne peut pas contenir cette information.

Utilisation

Avec l'ajout du nouveau lien entre le solveur de Siril et le catalogue local, et le nouveau lien entre le PCC de Siril et le catalogue local, une nouvelle commande `conesearch` a été créée (à partir de Siril 1.4.0) pour afficher les objets d'une image résolue. Pour afficher des étoiles qui contiennent des informations photométriques (l'indice B-V) et qui peuvent être utilisées pour la calibration, vous pouvez à présent utiliser :

```
conesearch -phot
```

C'est un bon moyen de vérifier que la résolution astrométrique et l'image sont alignées, en plus de la fonction d'annotation de l'objet (voir *annotations*).

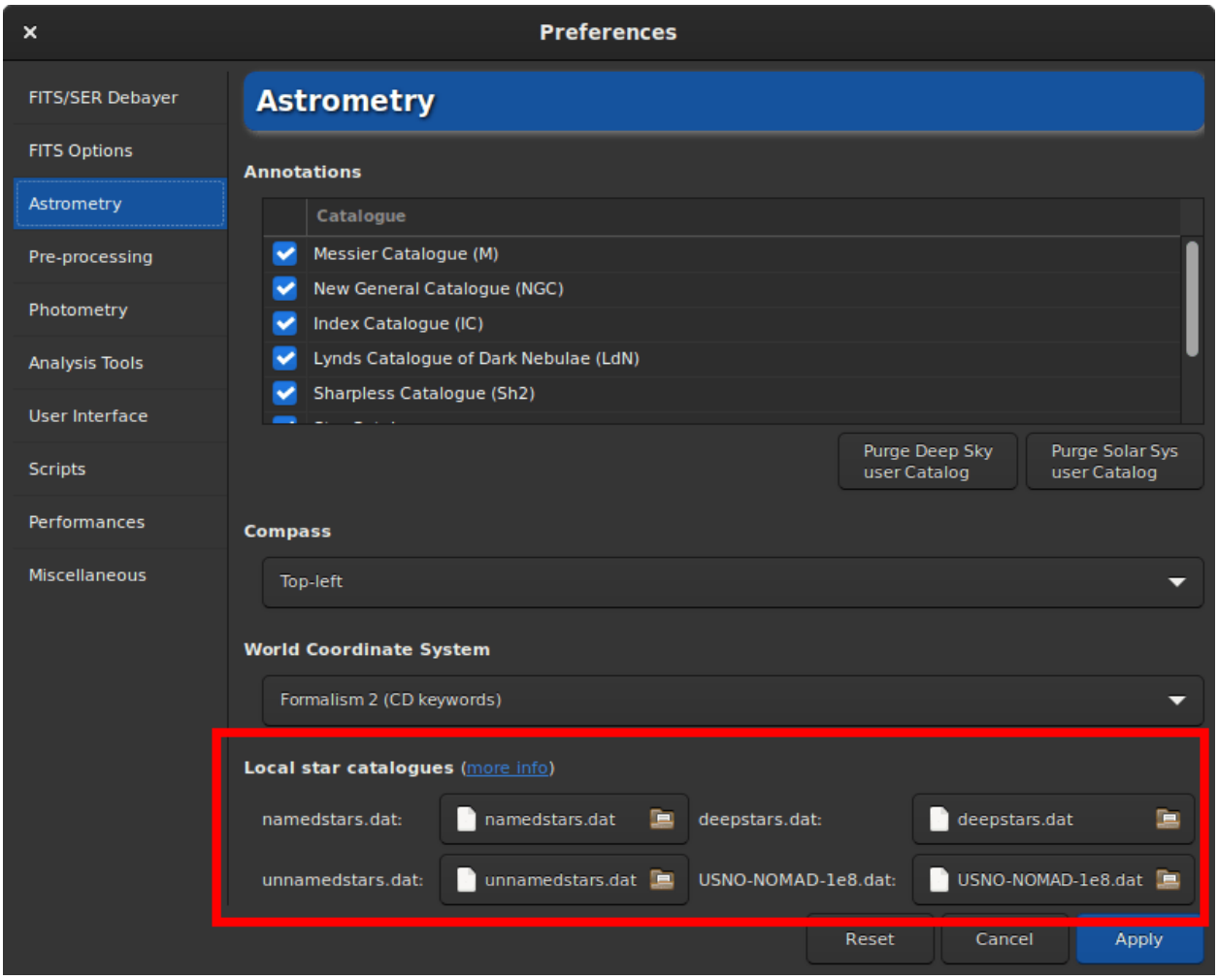


FIG. 4 – Préférences pour les catalogues locaux

Information technique

For photometry, Siril only uses the B-V index (or, for the Gaia catalogue, the effective temperature field Teff) 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.

Sommes Sha1 pour les 4 fichiers de catalogue :

```
4642698f4b7b5ea3bd3e9edc7c4df2e6ce9c9f7d  namedstars.dat
53a336a41f0f3949120e9662a465b60160c9d0f7  unnamedstars.dat
d32b78fd1a3f977fa853d829fc44ee0014c2ab53  deepstars.dat
12e663e04cae9e43fc4de62d6eb2c69905ea513f  USNO-NOMAD-1e8.dat
```

Licences pour les 4 fichiers de catalogue.

11.1.4 Utilisation du solveur local astrometry.net

Installation

Depuis la version 1.2, `solve-field`, le solveur de la suite astrometry.net, peut être utilisé par Siril pour résoudre astrométriquement des images ou des séquences d'images.

Pour les plates-formes Windows, le moyen le plus simple de l'obtenir est d'utiliser ansvr. Si vous n'avez pas changé le répertoire d'installation par défaut, c'est-à-dire *%LOCALAPPDATA%\cygwin_ansvr*, Siril le recherchera sans autre installation. Si vous avez cygwin et que vous avez construit astrometry.net à partir des sources, vous devez spécifier l'emplacement de la racine cygwin dans *Preferences*.

Pour MacOS, veuillez suivre ces instructions. Installez avec homebrew et ajoutez-le au `PATH`. Assurez-vous également que le programme fonctionne pour les images de test, comme indiqué dans la notice, et en dehors de Siril.

Pour les systèmes d'exploitation non Windows, l'exécutable devrait se trouver dans le `PATH`.

L'utilisation de cet outil permet de résoudre *à l'aveugle* des images, sans connaissance a priori de la zone du ciel qu'elles contiennent ou de sa résolution. C'est également une bonne alternative au solveur interne de Siril en cas d'échec, car il s'agit d'un outil dédié et éprouvé qui peut également prendre en compte la distorsion du champ.

Les paramètres par défaut devraient convenir, mais peuvent être modifiés si vous le souhaitez vraiment, en utilisant la commande *set* (valeurs par défaut spécifiées entre parenthèses) ou dans l'onglet *Astrometrie* des préférences. L'étendue de la gamme des échelles autorisées (15%), le rayon de recherche à partir des coordonnées initiales (10 degrés), l'ordre polynomial pour la distorsion du champ (0, désactivé), la suppression ou non des fichiers temporaires (oui), l'utilisation du résultat comme nouvelle distance focale et taille de pixel par défaut (oui).

Fichiers d'index

Astrometry.net a besoin de fichiers d'index pour fonctionner. Nous vous recommandons fortement d'utiliser les derniers fichiers d'index disponibles sur leur site web, c'est-à-dire les séries 4100 et 5200. Le champ de vision de chaque série est décrit dans leur page github. (la documentation officielle n'inclut pas encore ce tableau).

Sur un système Unix, vous pouvez simplement suivre les instructions de la documentation.

Sous Windows, si vous utilisez ansvr, ces fichiers d'index récents ne seront pas mis à disposition par l'Index Downloader. Vous pouvez toujours les télécharger séparément et les stocker là où sont conservés les autres fichiers d'index (il est recommandé de supprimer les anciens fichiers, bien que cela puisse perturber l'Index Downloader).

Comment cela marche

Tout comme le solveur interne, Siril va procéder à l'extraction des étoiles de vos images (afin de bénéficier du multithread interne) et soumettre cette liste d'étoiles à astrometry.net `solve-field`. Si vous souhaitez qu'astrometry.net explore l'index en multithread, vous devrez le spécifier dans le fichier `astrometry.cfg`.

11.1.5 Résoudre des séquences

Lorsqu'une séquence est chargée, un élément graphique supplémentaire est présent au bas de la boîte de dialogue.

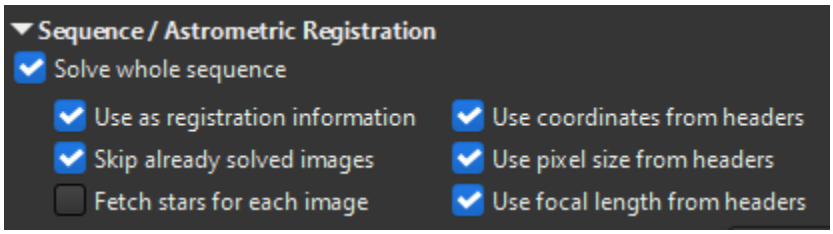


FIG. 5 – Options d'astrométrie de la séquence

You can specify to solve the whole sequence. The images already solved will be solved again unless the *Skip already solved images* box is ticked. To use *Astrometric registration*, you will need to process the whole sequence so that useful information is stored with the sequence (FWHM, number of stars, background level...), so leave the box *Use as registration information* ticked.

Lors de l'utilisation du solveur Siril avec des catalogues locaux ou lors de l'utilisation d'Astrometry.net, les informations contenues dans l'en-tête (si elles sont présentes) seront utilisées pour mettre à jour le centre cible de l'image et la résolution pour chaque image. Cependant, lors de l'utilisation du solveur Siril avec des catalogues en ligne, un seul catalogue d'étoiles sera téléchargé par défaut afin d'éviter trop de trafic sur le réseau et de requêtes sur le serveur. Si les images n'ont pas trop de dérive et ont la même résolution, cela est normalement suffisant. Cependant, si les images ne se chevauchent pas suffisamment ou si l'échantillonnage est différent, vous pouvez forcer le téléchargement d'un catalogue d'étoiles par image en cochant la case*Récupérer les étoiles pour chaque image*.

Finally, the 3 boxes on the right will enable to control if the platesolver should read target coordinates, pixel size and focal from each image header or the values specified in the dialog.

À la fin de la résolution de la séquence, les logs indiqueront combien d'images ont été résolues et si certaines ont été ignorées.

```
Sequence processing succeeded.
Execution time: 676.35 ms
3 images successfully platesolved out of 3 included
(2 were already solved and skipped)
```

La résolution de séquences est également disponible via la commande*seqplatesolve*.

Note

Lors de la résolution de séquences FITS ou d'un fichier*FITSEQ*, les images sont directement sauvegardées, sans création d'une nouvelle séquence. Pour les séquences FITS, si la séquence a été créée en utilisant des liens symboliques, les fichiers originaux ne sont pas mis à jour. Au lieu de cela, le nom du lien symbolique est utilisé pour créer un nouveau fichier FITS, en laissant l'original intact. Lors de la résolution d'une séquence*SER*, une nouvelle séquence avec le préfixe `ps_` est créée, car le SER ne peut pas stocker de données WCS.

11.1.6 Détection d'étoile

Par défaut, la détection des étoiles utilise l'algorithme *findstar* avec les paramètres actuels. Cela fonctionne très bien pour trouver de nombreuses étoiles, mais dans certains cas, nous aimerions détecter les étoiles manuellement, ou simplement voir celles qui sont utilisées. Une première étape serait d'ouvrir la fenêtre *PSF* et de lancer la détection d'étoiles, puis d'ajuster les paramètres (voir la *documentation* associée).

Une autre approche consisterait à sélectionner les étoiles une par une en les entourant d'une sélection puis, par un clic droit, à choisir *Pointer une étoile*. Plus il y a d'étoiles sélectionnées, plus l'algorithme a de chances de réussir.

Ensuite, dans la fenêtre d'astrométrie, développez la section de détection des étoiles et activez le bouton *Détection manuelle*. Au lieu de lancer *findstar*, il utilisera la liste actuelle des étoiles.

Note

Detection always uses the green layer for RGB images. For undebayered CFA images, the green layer is extracted on-the-fly and used instead.

11.1.7 Comprendre les résultats

Lorsqu'une solution astrométrique est trouvée, nous pouvons voir dans l'onglet Console ce type de messages :

```
Up is -5.26 deg CounterClockWise wrt. N
Resolution:      3.051 arcsec/px
Focal length:  254.21 mm
Pixel size:      3.76 µm
Field of view:    04d 51m 58.27s x 03d 01m 1.21s
Image center: alpha: 21h02m02s, delta: +68°10'48"
Was 119.64 arcmin from initial value
Saved focal length 254.21 and pixel size 3.76 as default values
Flipping image and updating astrometry data.
```

La solution astrométrique nous donne les coordonnées équatoriales J2000 du centre de l'image, la dimension horizontale et verticale projetée de l'image sur le ciel, la longueur focale qui pourrait donner ce champ pour la taille de pixel donnée et par conséquent l'échantillonnage réel de l'image, l'angle que fait l'image avec l'axe nord, le champ de vision et le centre de l'image. Il indique également la distance par rapport au centre initial spécifié.

En cas d'échec, vérifiez que les coordonnées de départ et la taille des pixels sont correctes et essayez de changer la magnitude, cela changera la quantité d'étoiles téléchargées à partir des catalogues, et peut-être que plus d'étoiles seront identifiées. Si la résolution astrométrique de Siril ne trouve pas de solution, il est toujours possible d'utiliser un outil externe pour le faire, la solution sera écrite dans l'en-tête*FITS* d'une manière ou d'une autre.

11.1.8 Visualiser les distorsions

Pour vérifier la validité de la solution, vous pouvez utiliser la commande *conesearch*. Celle-ci devrait afficher les positions des étoiles trouvées dans les catalogues et vérifier si elles correspondent aux étoiles réelles de l'image.

Les deux images ci-dessous montrent des annotations dans le coin supérieur droit d'une image présentant une distorsion importante. L'image du haut est la solution linéaire, tandis que celle du bas a été résolue en tenant compte des polynômes cubiques.

If platesolving with distortions is sucessfull, the *disto* command or the *Tools* → *Image Analysis* → *Show Distortions* menu can display a representation of the corrections as an overlay on the image.

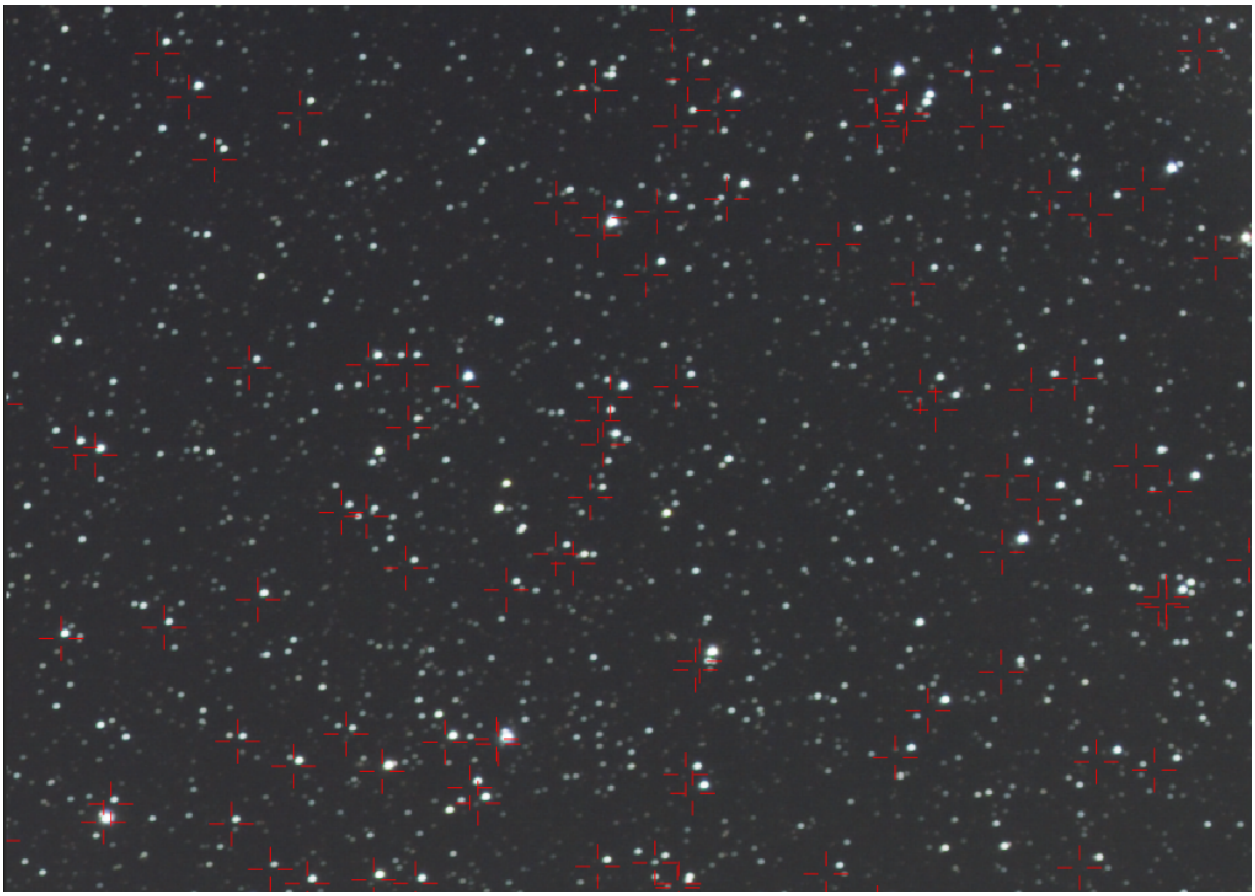


FIG. 6 – Annotations d'étoiles avec solution linéaire

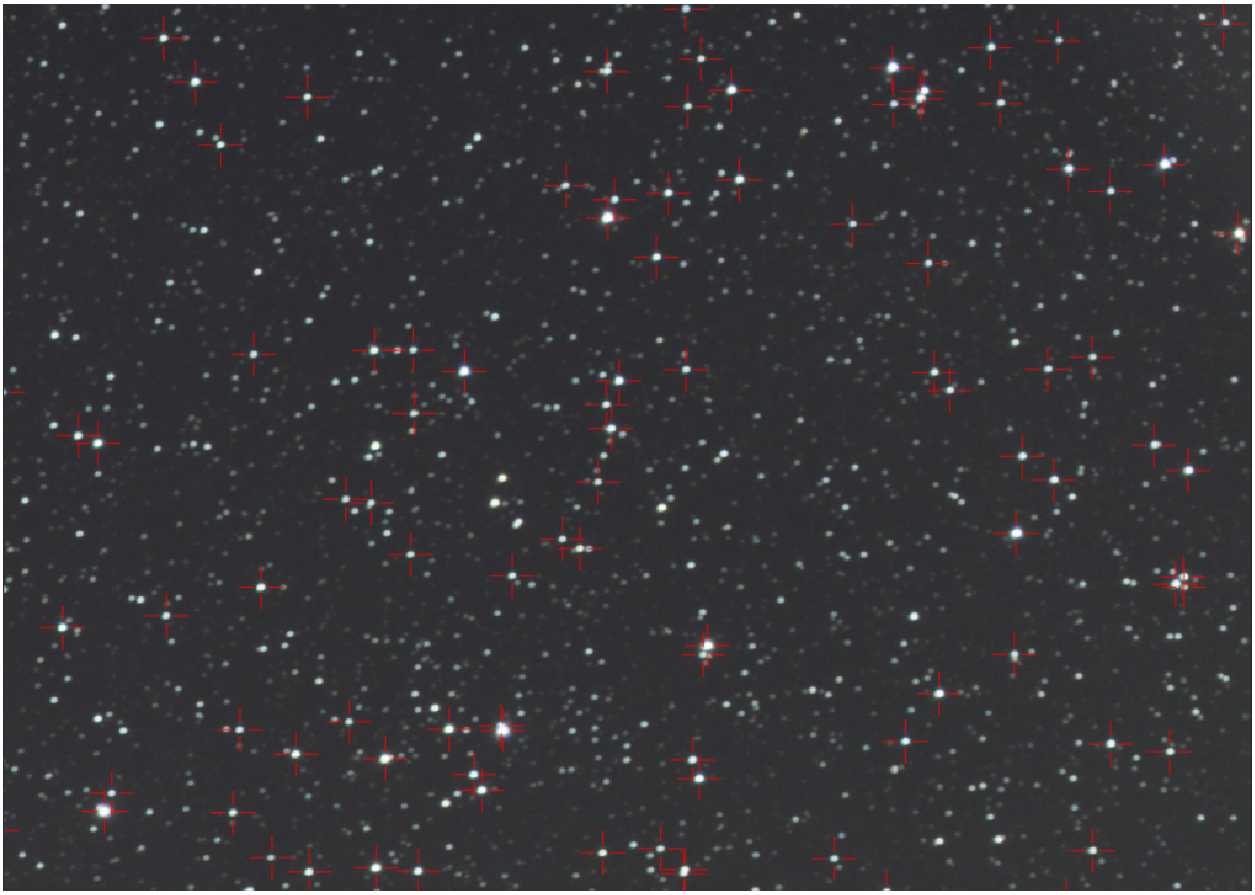


FIG. 7 – Annotations d'étoiles avec solution cubique

Ligne de commande Siril

```
disto [clear]
```

Montre le champ de distorsion sur une image résolue dont la solution comprend des termes de distorsion

Passer l'option **clear** pour désactiver

Un exemple est donné ci-dessous.

FIG. 8 – Image avec superposition de distorsion

Scripts Sirilpy

If you want to visualize the distortion field in 3D, you can use the Python script `Distortion3D.py` in the *Scripts->Python Scripts* menu. You will need to enable getting scripts from the *scripts repository*.

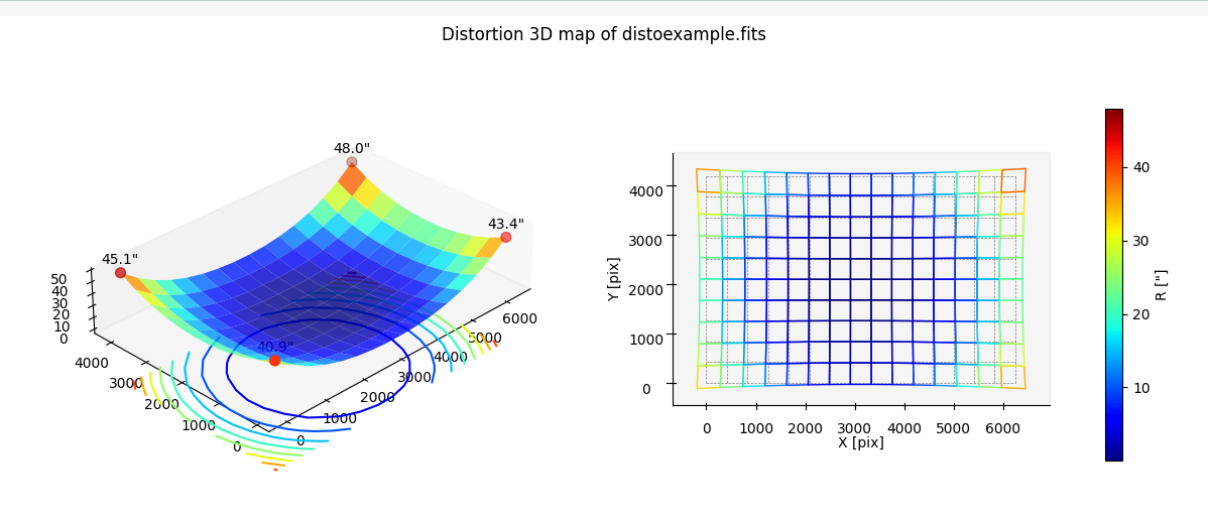


FIG. 9 – Carte de distorsion 3D

11.2 Annotations

Les annotations sont des glyphes affichés au-dessus des images pour indiquer la présence d'objets célestes connus, comme les galaxies, les étoiles brillantes, etc. Elles proviennent de catalogues, mais ne peuvent être affichées que sur des images pour lesquelles nous savons dans quelle partie du ciel elles sont, des images qui ont été **résolues par astrométrie** et qui contiennent les informations du système de coordonnées mondiales (WCS) dans leur en-tête, donc uniquement des fichiers *FITS* ou *Astro-TIFF*.

La résolution astrométrique peut être effectuée dans Siril dans l'entrée *Outils* → *Astrométrie* → *Résolution astrométrique...*, ou en utilisant des outils externes comme astrometry.net ou ASTAP.

Lorsqu'une image résolue astrométriquement est chargée dans Siril, vous pouvez voir les coordonnées dans le ciel pour le pixel sous le pointeur de la souris affiché dans le coin inférieur droit et les boutons relatifs aux annotations deviennent disponibles. Le premier bouton active les annotations des objets, le second la grille céleste et la boussole.

11.2.1 Catalogues d'annotations hors ligne

Siril vient avec une liste de catalogue prédéfini pour les annotations :

- Catalogue de Messier (M)
- New General Catalogue (NGC)
- Index Catalogue (IC)
- Lynds Catalogue of Dark Nebulae (LdN)
- Catalogue Sharpless (Sh2)
- Star Catalogue (3661 étoiles les plus brillantes)
- Lignes des constellations de l'UAI
- IAU constellations names (positions collected from this page)

En supplément, 2 *catalogues utilisateurs* peuvent être utilisés :

- Catalogue d'objets du ciel profond de l'utilisateur (DSO)
- Catalogue d'objets du système solaire pour l'utilisateur (SSO)

Elles sont alimentées par les commandes décrites dans la section *recherche d'un objet connu*.

Ils sont stockés dans le répertoire des paramètres de l'utilisateur. Leur emplacement dépend du système d'exploitation :

- Pour les OS *Unix-based* ils seront dans `~/.config/siril/catalogue`
- sur *Windows* il sont dans `%LOCALAPPDATA%\siril\catalogue`.

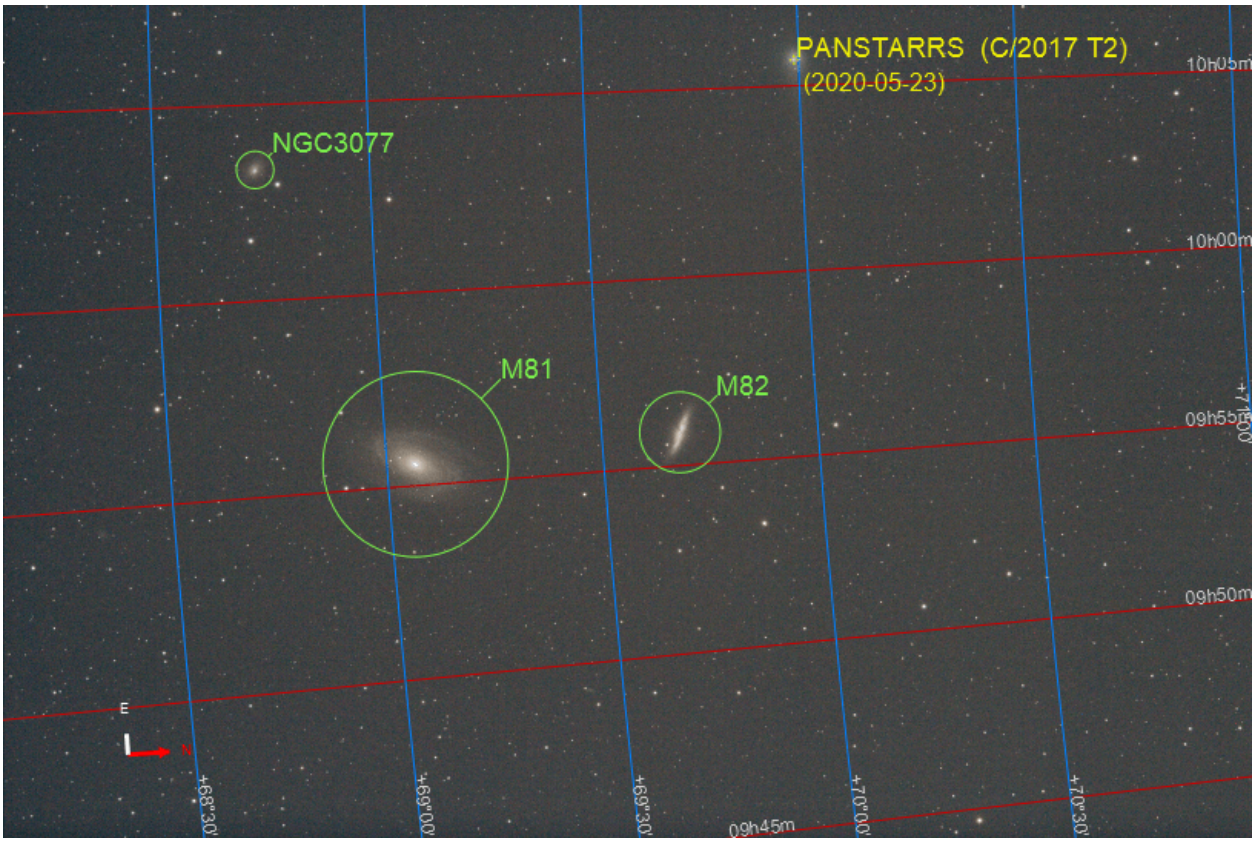


FIG. 10 – Vue de l'image complète annotée

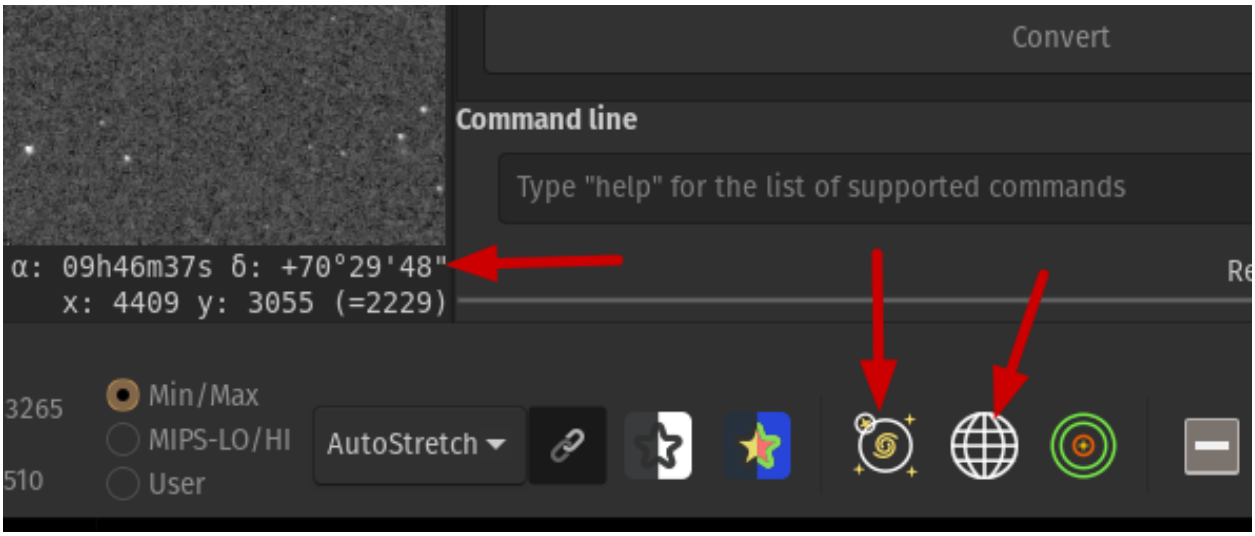


FIG. 11 – Bouton pour annotations

Tous les catalogues ci-dessus peuvent être activés/désactivés dans le menu *Preferences* → *Astrometrie tab*.

Les deux *catalogues définis par l'utilisateur* peuvent également être purgés (*c'est-à-dire* supprimés) à l'aide des boutons appropriés.

Un curseur sur le côté droit vous permet de naviguer facilement dans la liste de catalogues.

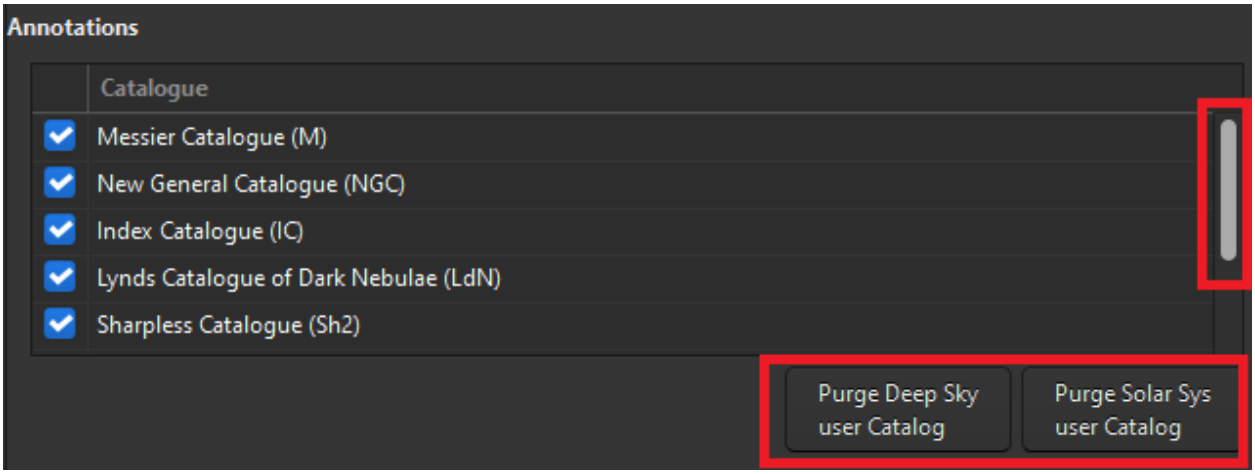


Fig. 12 – Gestion des catalogues dans Préférences/Astrométrie

There's also a simpler way of accessing this list, as well as the button for displaying/hiding catalog annotations. Right-click on the annotations button in the toolbar and the list will be displayed in a popup window.

Ces catalogues d'annotations sont utilisés principalement à des fins d'affichage. À partir de Siril 1.3, ils sont également utilisés pour localiser le centre de l'image pour l'outil d'astrométrie. Si l'objet est trouvé localement, le résolveur sera affiché comme Local. Si ce n'est pas le cas, un résolveur en ligne sera utilisé.

11.2.2 Catalogues d'annotations en ligne

Vous pouvez interroger d'autres bases de données que celles déjà fournies avec Siril, décrites dans la section *catalogues d'annotations hors ligne*. Cela ne fonctionne, encore une fois, que pour les images résolues par astrométrie.

Starting from Siril 1.3, this is possible with the command *conesearch*. This new command replaces and expands capabilities previously provided by `nomad` and `solsys` from Siril 1.2.

Ligne de commande Siril

```
conesearch [limit_magnitude] [-cat=] [-phot] [-obscode=] [-tag={on|off}] [-log=
↪{on|off}] [-trix=] [-out=]
```

Affiche les étoiles du catalogue local par défaut pour l'image chargée résolue, jusqu'à la magnitude limite **limit_magnitude** fournie (13 par défaut pour la plupart des catalogues, sauf 14.5 pour aavso_chart, 20 pour solsys, et omis pour pgc).
Un autre catalogue en ligne peut être spécifié avec **-cat=**, prenant les valeurs
- for stars : tycho2, nomad, gaia, localgaia, ppmxl, bsc, apass, gcvs, vsx, simbad, aavso_chart
- for exoplanets : exo
- for deep-sky : pgc
- for solar system objects : solsys (closest IAU observatory code can be passed with the argument **-obscode=** for better position accuracy)

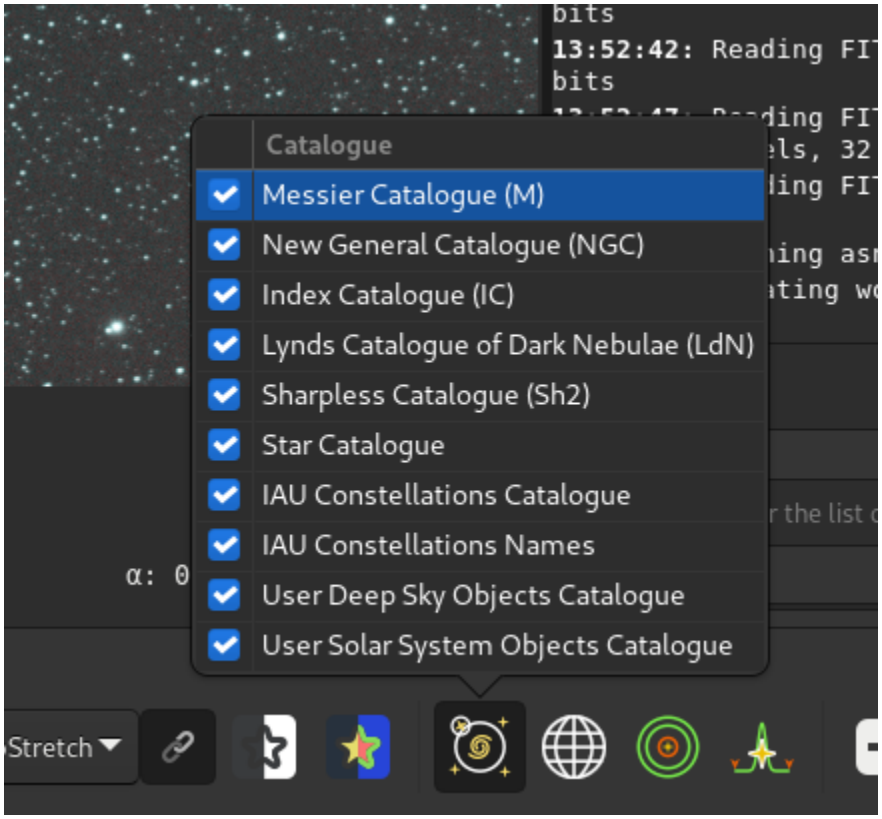


FIG. 13 – Catalogue management in the pop-up window triggered by right-clicking on the annotation button.

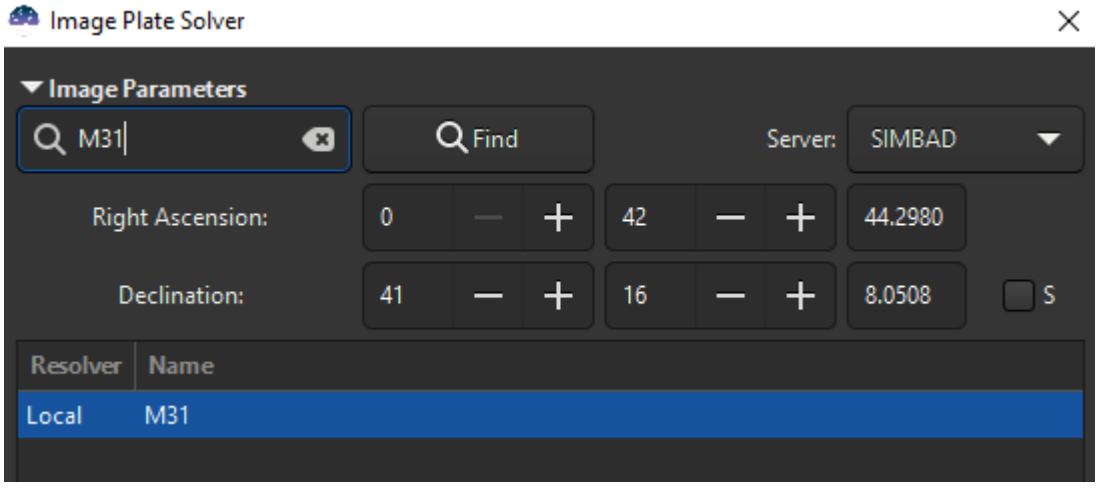


FIG. 14 – Objet résolu à partir des catalogues d'annotations locaux

Pour les catalogues d'étoiles contenant des données photométriques, les étoiles sans information B-V seront conservées ; elles peuvent être exclues en passant l'option **-phot**
L'argument **-trix=** peut être passé à la place d'un catalogue suivi d'un nombre entre 0 et 511 pour tracer les étoiles contenues dans les catalogues locaux trixel de niveau 3 (pour l'usage des développeurs principalement)

Certains catalogues (bsc, gcvs, pgc, exo, aavso_chart, varisum et solsys) affichent également, par défaut, les noms à côté des marqueurs dans l'écran (interface graphique uniquement) et les listent dans le journal. Pour d'autres qui ont un plus grand nombre d'objets, à savoir vsx et simbad, l'information peut aussi être affichée, mais, comme elle risque d'encombrer l'écran, elle n'est pas activée par défaut. Ce comportement peut être activé/désactivé avec les options **-tag=on|off** pour afficher les noms à côté des marqueurs et **-log=on|off** pour lister les objets dans le journal de la console

La liste des éléments présents dans l'image peut éventuellement être sauvegardée dans un fichier csv en passant l'argument **-out=**

This command is accessible from the graphical interface via the *Tools* → *Astrometry* → *Annotate...*.

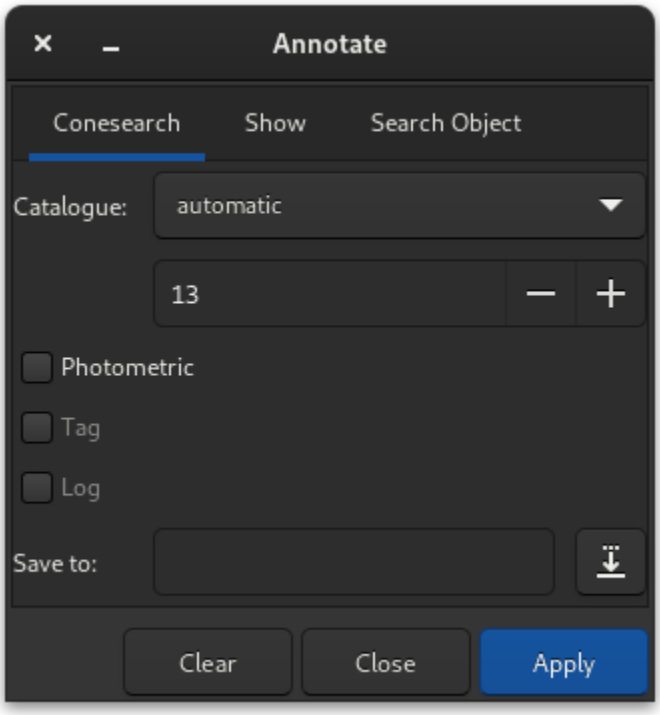


Fig. 15 – Version graphique de la commande *conesearch*

Le tableau ci-dessous répertorie tous les catalogues disponibles, ainsi que les liens vers les données originales.

Tableau 1 – Tableau des catalogues disponibles

Catalogue	*-cat=*	Référence
Local	*Omis*	en utilisant *local star catalogues*
Tycho2	tycho2	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/297
Nomad	nomad	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/297
GAIA DR3	gaia	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/355
PPMXL	ppmxl	https://cdsarc.cds.unistra.fr/viz-bin/cat/I/317
Catalogue Bright Stars	bsc	https://cdsarc.cds.unistra.fr/viz-bin/cat/V/50
APASS DR9	apass	https://cdsarc.cds.unistra.fr/viz-bin/cat/II/336
GCVS	gcvs	https://cdsarc.cds.unistra.fr/viz-bin/cat/B/gcvs
VSX	vsx	https://cdsarc.cds.unistra.fr/viz-bin/cat/B/vsx
SIMBAD	simbad	https://simbad.cds.unistra.fr/simbad/sim-tap/
PGC/HYPERLEDA	pgc	https://cdsarc.cds.unistra.fr/viz-bin/cat/VII/237
Archives des exoplanètes	exo	https://exoplanetarchive.ipac.caltech.edu
AAVSO VSP	aavso_chart	https://app.aavso.org/vsp/
skybot Système solaire IMCCE	solsys	https://vo.imcce.fr/webservices/skybot/?conesearch

Les requêtes suivantes vous sont fournies grâce à :

- Vizier (par l'intermédiaire du CDS Vizier TAP query service)
- Simbad (par l'intermédiaire du CDS Simbad TAP query service)
- IMCCE (par l'intermédiaire de son service skybot conesearch)
- Archives de la NASA sur les exoplanètes (via son service de recherche TAP)

For solar system object queries, you may pass an additional parameter **-obscode=**, the 3-symbol code for an IAU observatory close by to your observing location. This will improve annotations accuracy. Please note that results may still slightly differ from those obtained by making a *direct ephemerides query for a specific object*, which uses the exact observing location (if present in the FITS header).

Astuce

A preferred observatory code may be set in the *Astrometry* tab in the *Preferences* dialog. If set, this will be used for the *Solar System Objects* GUI tool and will also be used in the *conesearch* command unless the **-obscode=** argument is provided.

These additional annotations will be displayed in **RED**, to differentiate them from offline annotations, shown in **GREEN**. These annotations will be erased as soon as the `Show Objects names` button is toggled.

11.2.3 Catalogues supplémentaires

You may want to display your own user catalogues. This can be done with the command *show*. This command can also be used to display, for instance, csv files created with the feature to find *comparison stars*.

Ligne de commande Siril

```
show [-clear] [{ -list=file.csv | [name] RA Dec }] [-nolog] [-notag]
```

Affiche un point sur l'image chargée et résolue en utilisant le catalogue temporaire d'annotations de l'utilisateur, basé sur ses coordonnées équatoriales. L'option **-clear** efface d'abord ce catalogue et peut être utilisée seule.

Plusieurs points peuvent être transmis en utilisant un fichier CSV avec l'option **-list=** contenant au moins les colonnes ra et dec. Si le fichier transmis contient également un nom de colonne, les noms seront utilisés comme balises dans l'image et listés dans la console, à moins qu'ils ne soient désactivés avec les options **-notag** et **-nolog**.

Cette fonction n'est disponible qu'à partir de l'interface graphique de Siril

Ces catalogues peuvent être des fichiers *csv* (séparés par des virgules), respectant les règles suivantes :

- les lignes de commentaires, s'il y en a, doivent commencer par un signe #
- une ligne doit être présente en haut avec les noms des colonnes, séparés par des virgules
- au moins les colonnes *ra* et *dec* doivent être fournies, en degrés décimaux.
- les colonnes peuvent être rédigées sans ordre particulier
- d'autres colonnes peuvent être fournies (en anglais) :
  - *name* (str), pour le nom
  - *diameter* (double), le diamètre de l'objet en arcmin
  - *mag* (double), la magnitude de l'objet
  - *type* (str), qui sera ajouté entre *()* après le nom dans la Console

D'autres colonnes que celles énumérées ci-dessus peuvent être transmises, mais elles ne seront pas utilisées.

This command is accessible from the graphical interface via the *Tools* → *Astrometry* → *Annotate...*. You can either load a `*.csv` file containing several lines of comma-separated ra, dec coordinates in *Import CSV*, or a single point in the *Single Coordinates* tab.

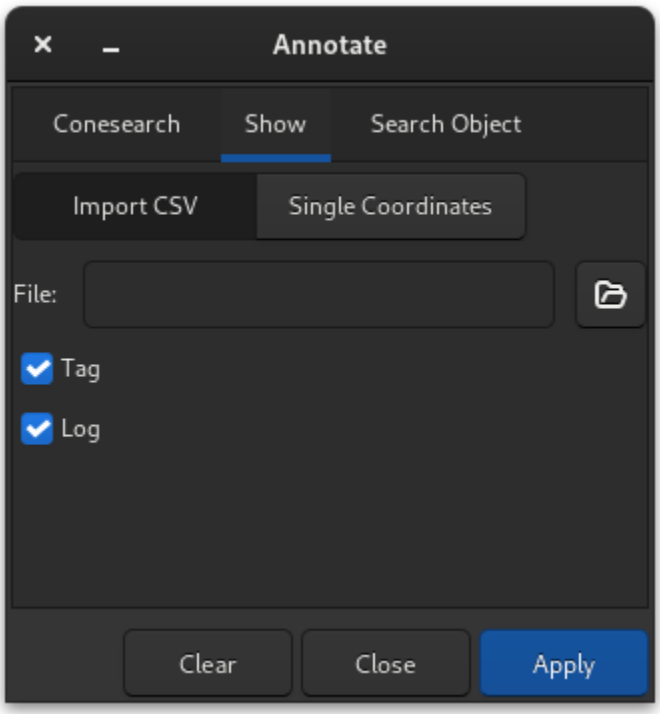


FIG. 16 – Version graphique de la commande *show*

In the case of a single point, you can save the result in the *Deep Sky user catalogue* by clicking on the save button as shown in the illustration bellow :

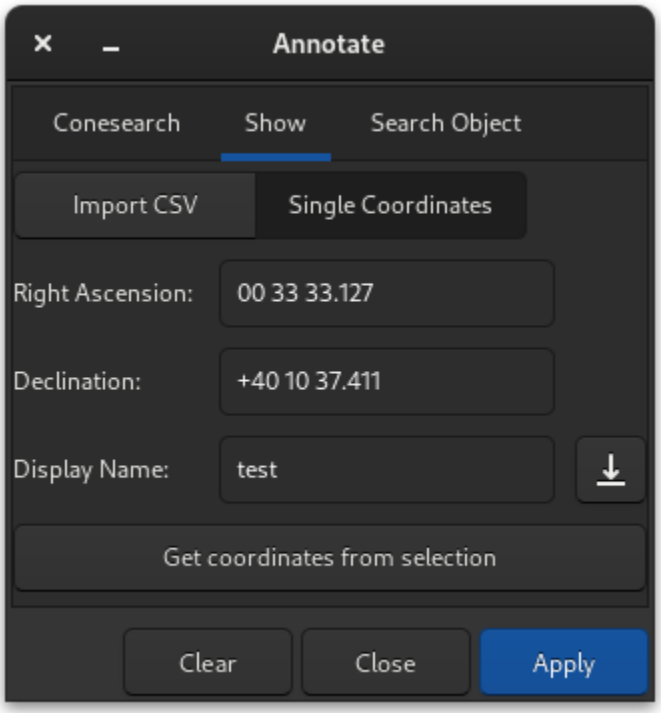


FIG. 17 – It is possible to show only one single point and to save it in the *Deep Sky user catalogue*.

Astuce

The save button will only work if the field *Display Name* is filled.

Astuce

You can select a star or object and obtain its coordinates using the *Get coordinates from selection* button.

Liste des catalogues d'utilisateurs connus :

Parfois, les utilisateurs créent leurs propres catalogues, nous pouvons essayer de mettre un lien ici pour aider tout le monde.

— Étoiles variables, extraites de GCVS 5.1, discuté ici en français, (`lien`).

Avertissement

Contrairement aux instructions discutées dans le sujet lié, il n'est pas recommandé de remplacer le catalogue Ciel Profond utilisateur par de tels fichiers. Leur utilisation est déconseillée car certains d'entre eux peuvent être particulièrement volumineux et ralentiraient considérablement chaque affichage d'annotation.

11.2.4 Recherche d'un objet connu

Si vous savez qu'un objet spécifique se trouve quelque part dans l'image (sinon, voir la section *recherche d'un objet inconnu*), il est possible de l'ajouter aux annotations.

Objets du ciel profond

Load a plate resolved image and type `Ctrl+Shift+/` or *Tools* → *Astrometry* → *Annotate...* then go to the *Search Object* tab of the dialog box.

A search dialog box is available, as illustrated below, in which object names can be entered.

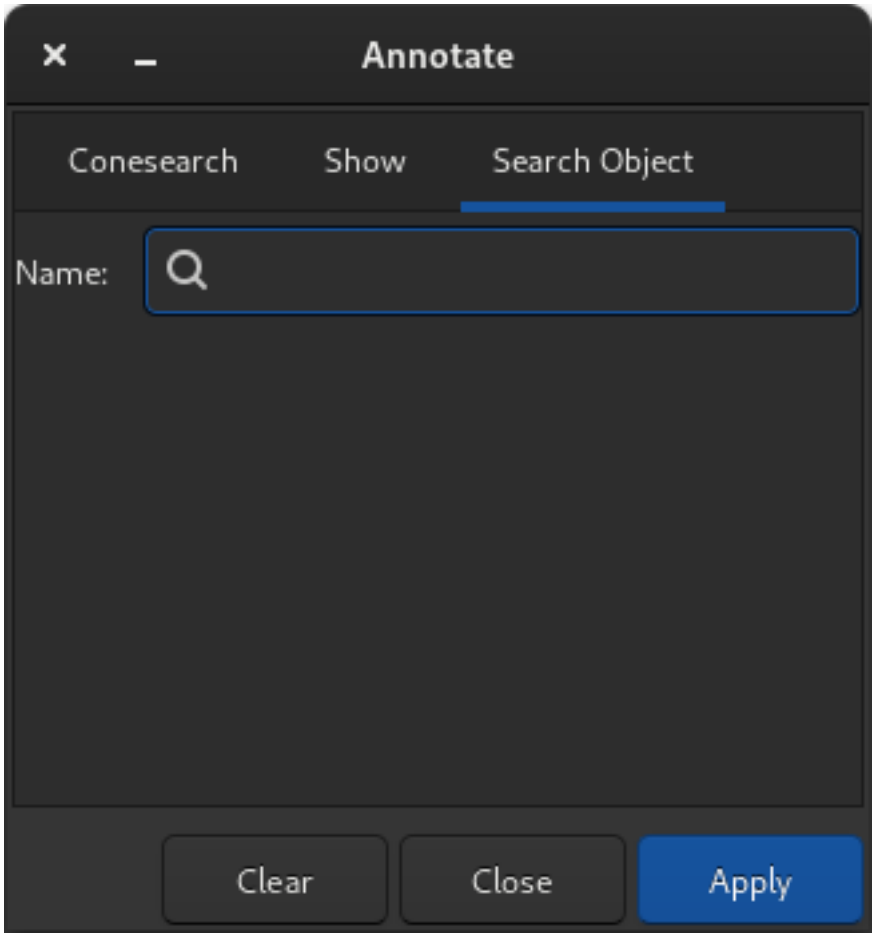


FIG. 18 – Boîte de dialogue Rechercher un objet

Pressing `Enter` or *Apply* will first search for this name the existing annotation catalogues in case it already exists under another name. If not it will send an online request to SIMBAD 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**.

FIG. 19 – Objets du ciel profond à partir de catalogues d'utilisateurs et de catalogues prédéfinis

Exemples d'entrées valides (non sensible à la case) :

— `HD 86574` ou `HD86574` sont tout deux valide pour cette étoile

Objets du système solaire

From Siril version 1.2, objects from the solar system can also be searched for, using the Miriade ephemcc service. This is done in the same manner as for Deep Sky Objects, but 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, `dp:` for dwarf planets and `s:` for natural satellites. If you query an image taken from a close enough date and time (same night) than another image already annotated with SSOs, their cached positions will be used and corrected by each object velocity as returned by the ephemerids. The items of this catalogue are displayed in **YELLOW**.

Exemples d'entrées valide (non sensible à la case) :

— `c:67p` ou `c:C/2017 T2` sont une forme valide pour les comètes
— `a:1` et `a:ceres` sont tous deux valides pour *(1) Ceres*
— `a:2000 BY4` est valide pour *103516 2000 BY4*
— `p:4` or `p:mars` sont tous les deux valides pour Mars
— `dp:Pluto` est valide pour Pluton
— `s:Moon` ou `s:Io` est valide pour les satellites naturels.

Avertissement

Les images qui n'ont pas de clé d'en-tête `DATE-OBS` ne peuvent pas être annotées pour les objets du système solaire. Les images qui n'ont pas d'informations sur la position de l'observateur (clés d'en-tête SITELAT, SITELONG et SITEELEV) seront tout de même annotées, mais en supposant une position géocentrique de l'observateur, c'est-à-dire comme s'il observait depuis le centre de la Terre. En fonction de la distance des objets par rapport à la Terre, les positions peuvent être légèrement décalées par rapport à leur position réelle.

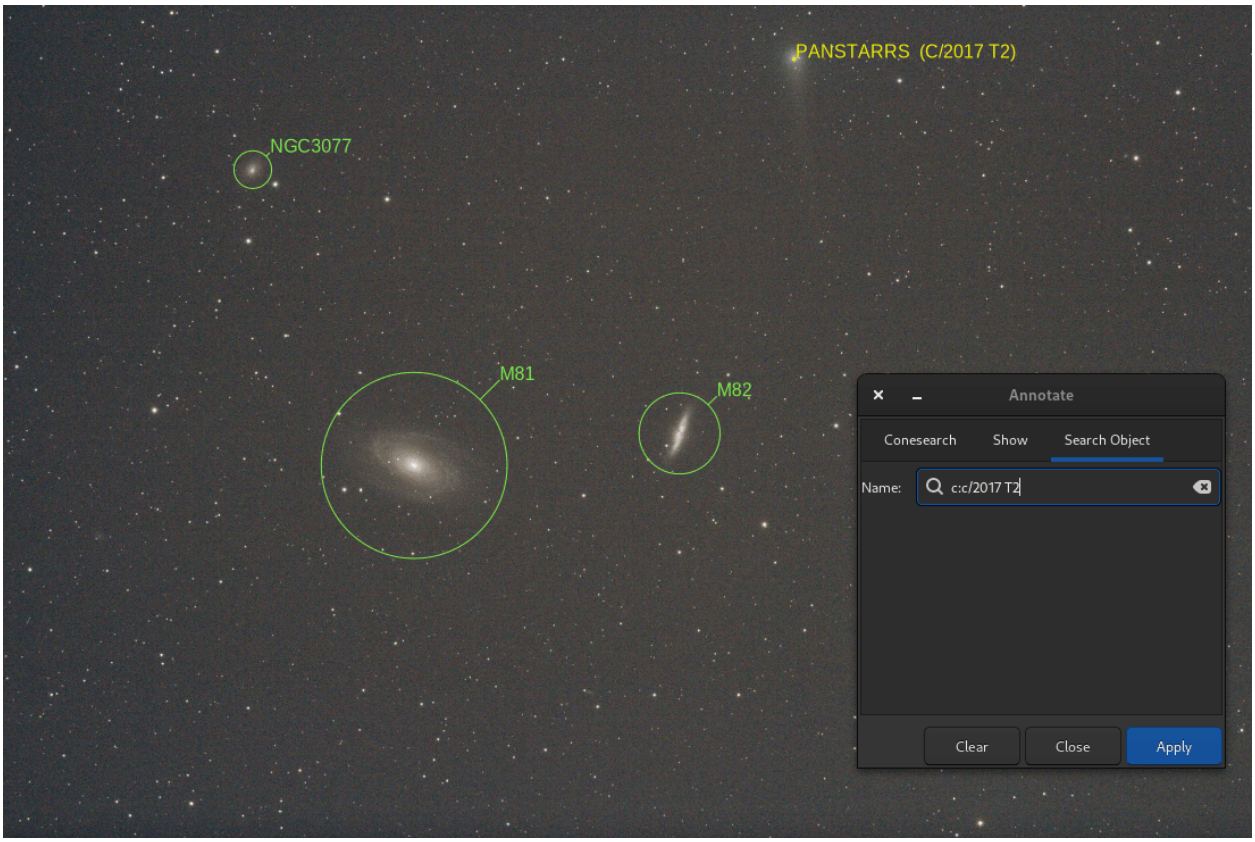


Fig. 20 – Résultat d'un processus de recherche dans le système solaire

Commande *catsearch*

La même fonctionnalité est accessible par la commande catsearch <catsearch>` :

Ligne de commande Siril

```
catsearch name
```

Recherche un objet par **nom** et l'ajoute au catalogue d'annotations de l'utilisateur. L'objet est d'abord recherché dans les catalogues d'annotations, s'il n'est pas trouvé, une demande est faite à SIMBAD.

L'objet peut être un objet du système solaire, auquel cas un préfixe, "a :" pour astéroïde, "p :" pour planète, "c :" pour comète, "dp :" pour planète naine ou "s :" pour satellite naturel, doit précéder le nom de l'objet. La recherche

est effectuée pour la date, l'heure et le lieu d'observation figurant dans l'en-tête de l'image, à l'aide du service Miriade de l'IMCCE

11.2.5 Recherche d'un objet inconnu

Particulièrement utile pour les travaux de photométrie, il est possible d'identifier une étoile ou d'autres objets dans l'image en dessinant une sélection autour d'eux, en faisant un clic droit pour faire apparaître le menu contextuel, et en sélectionnant l'entrée *PSF*.

La fenêtre PSF s'ouvre et, s'il s'agit d'une étoile, elle affiche les paramètres de l'ajustement gaussien, mais aussi un lien Web en bas à gauche de la fenêtre. En l'ouvrant, vous accéderez à la page SIMBAD pour les coordonnées de l'objet et, dans de nombreux cas, vous obtiendrez le nom de l'objet.

SIMBAD ne contient pas tous les objets connus, mais les coordonnées de la page peuvent être utilisées comme point de départ pour rechercher l'objet dans d'autres catalogues en ligne, par exemple Gaia DR3 (VizieR).

CHAPITRE 12

PSF Dynamique

Cette section décrit les deux étapes essentielles qui ont pour but de détecter les étoiles dans les images individuelles. La détection sur une seule image peut être lancée ou affinée à l'aide de *Outils Analyse de l'image* → *PSF dynamique* ou avec le raccourci `Ctrl` + `F6`.

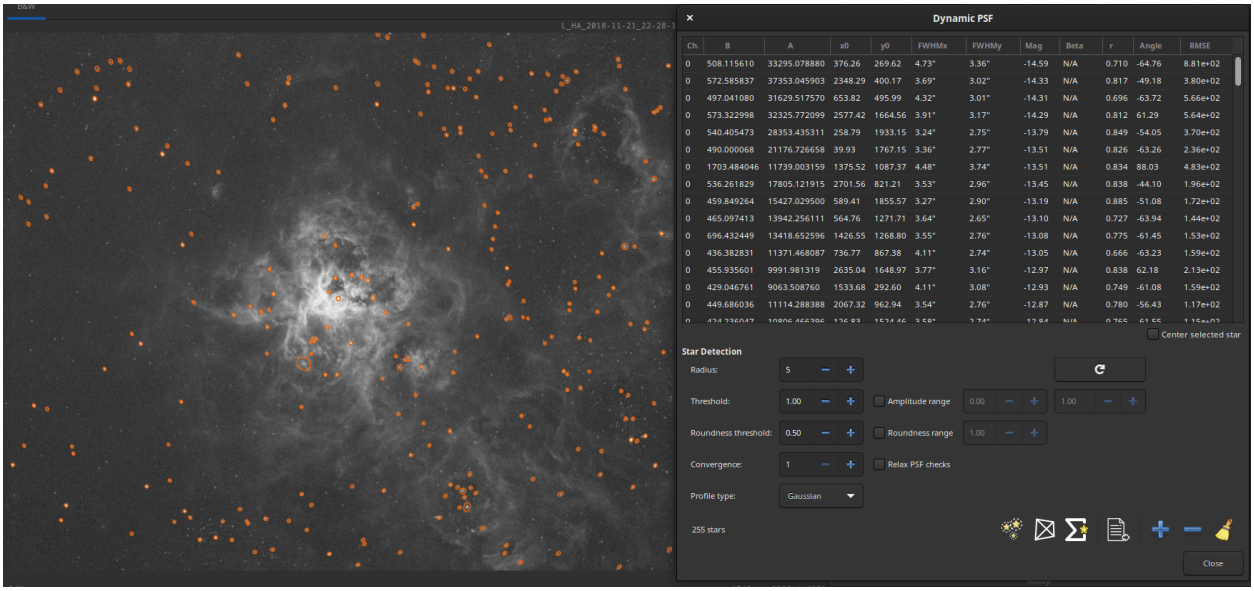


FIG. 1 – PSF dynamique sur une image de ciel profond.

Le processus est le suivant :

- réaliser une première détection des étoiles candidates potentielles
- Ajuste un modèle PSF sur chaque candidat. Exécute des contrôles d'intégrité basés sur les paramètres d'ajustement du modèle pour vous assurer qu'il s'agit bien d'une étoile et rejetez les candidats qui ne le sont pas.

À la fin de ce processus, le résultat est une liste d'étoiles, avec les positions dans l'image par rapport au coin supérieur gauche et les quantités mesurées de toutes les étoiles de la liste.

12.1 Candidat étoiles initiale

Si l'emplacement des étoiles peut sembler évident lorsqu'on regarde une image, il est un peu plus difficile de traduire ce processus en opérations et critères mathématiques. Cette section décrit brièvement l'algorithme sous-jacent. Il est inspiré du manuel du logiciel DAOPHOT [Stetson1987], avec des simplifications apportées pour améliorer les performances. L'algorithme original visait à être exhaustif dans la détection de toutes les étoiles possibles, dans le but de construire des catalogues d'étoiles, alors que Siril a besoin de détecter des étoiles principalement comme élément pour l'alignement . Il doit également être résistant à la grande variété d'images qui lui sont soumises - la plupart d'entre nous n'ont pas un équipement astronomique de qualité professionnelle dans leur jardin - et nous avons dû faire des choix concernant la connaissance préalable des conditions d'imagerie (échantillonnage, qualité du ciel, etc.).

Au fil des ans, notre implémentation a évolué pour aboutir à ce qu'elle est aujourd'hui. Elle vise à ne pas manquer les étoiles très brillantes, qui sont importantes pour l'alignement, et à rejeter autant que possible les valeurs aberrantes, tout en restant raisonnablement rapide pour le faire.

Ceci peut être décomposée selon les étapes suivantes :

— calcule les statistiques de l'image pour le niveau du fond, comme la médiane de l'image, et son bruit. Cela suppose que l'image est relativement uniforme. Par conséquent, la détection aura tendance à être moins efficace dans les coins si un vignettage important est présent après le pré-traitement.
— Calculez également une plage dynamique, définie comme le maximum de l'image moins son arrière-plan. Cela sera utile par la suite pour détecter les étoiles saturées.
— lisser l'image avec un noyau gaussien. Le lissage idéal serait d'utiliser le noyau qui a la même FWHM que l'image. Au lieu de cela, nous avons choisi une taille arbitraire qui a donné des résultats satisfaisants dans une très large gamme de conditions. Cela permet d'être "aveugle" aux paramètres de l'imagerie.
— sur la version lissée de l'image, détecte les maxima locaux sur un niveau défini comme le niveau du fond plus X fois le bruit (X peut être modifié en utilisant la valeur `threshold` dans l'interface graphique). Assurez-vous que ce soit un maximum sur une certaine zone (définie par le paramètre `radius`).
— Exécution d'un contrôle d'intégrité pour s'assurer que le maximum et ses voisins sont bien au-dessus des pixels environnants (pour rejeter les taches dans les parties lumineuses de la nébuleuse par exemple).
— exécute un contrôle d'intégrité pour déterminer si le cœur au centre des maxima est saturé, c'est-à-dire s'il est proche de la limite supérieure de la plage dynamique. Si c'est le cas, exécute un algorithme pour détecter la limite de la partie saturée.
— Utilise les dérivées premières et secondes le long d'une ligne horizontale et verticale passant par le centre pour calculer le fond du ciel autour de l'étoile, son amplitude et sa largeur dans toutes les directions (haut, bas, gauche et droite).
— Si les paramètres sont suffisamment symétriques dans toutes les directions (jusqu'au paramètre `rondeur`), confirmez l'étoile comme un candidat potentiel.

Une fois que la liste des candidats potentiels a été définie, ils sont triés par amplitudes décroissantes et introduits dans l'algorithme d'ajustement du PSF décrit dans la section *minimisation*.

12.2 Modèles

Deux modèles sont utilisés dans la fenêtre Dynamic PSF. En général, le modèle Moffat est beaucoup plus adapté à l'ajustement d'objets tels que les étoiles.

Théorie

1. Une fonction d'ajustement gaussienne elliptique définie comme suit

$$G(x,y) = B + Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)}. \tag{12.1}$$

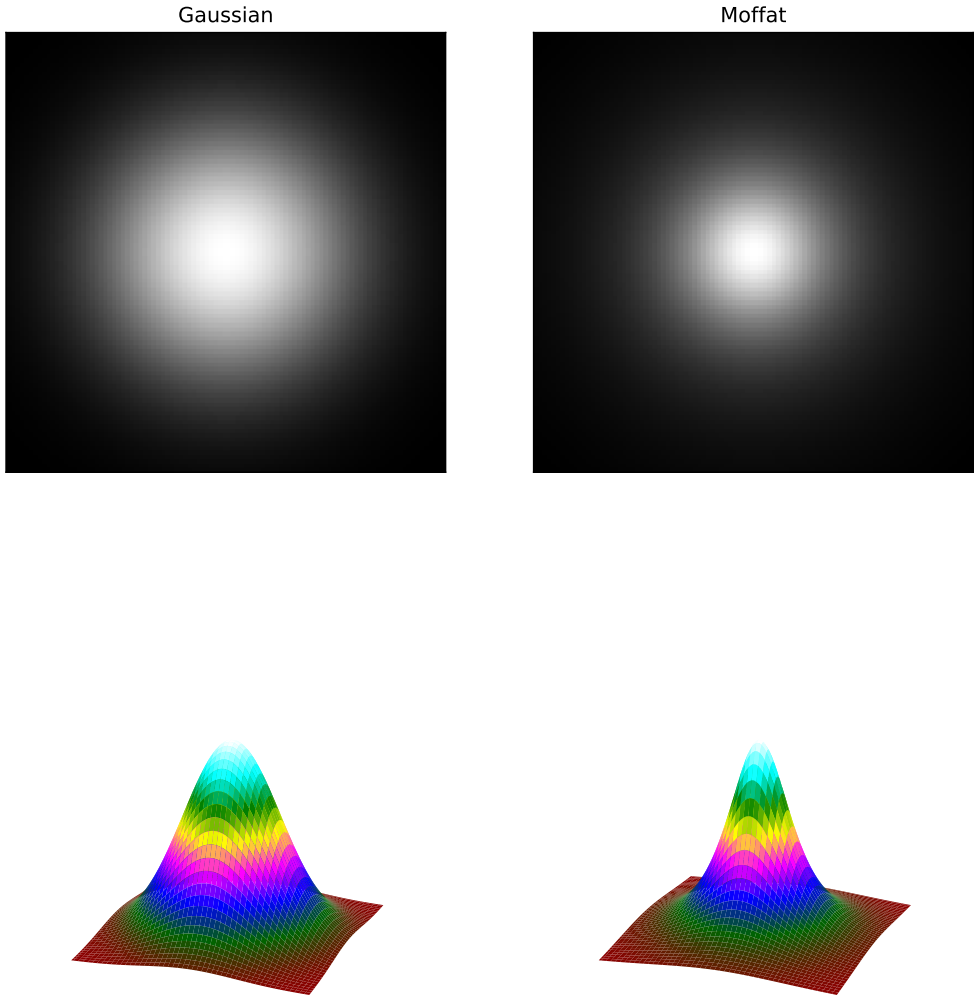


FIG. 2 – Affichage de deux PSFs circulaires selon un profil gaussien et un profil de Moffat. Les deux modèles utilisent les mêmes paramètres et le profil de Moffat utilise $\beta = 1.4$.

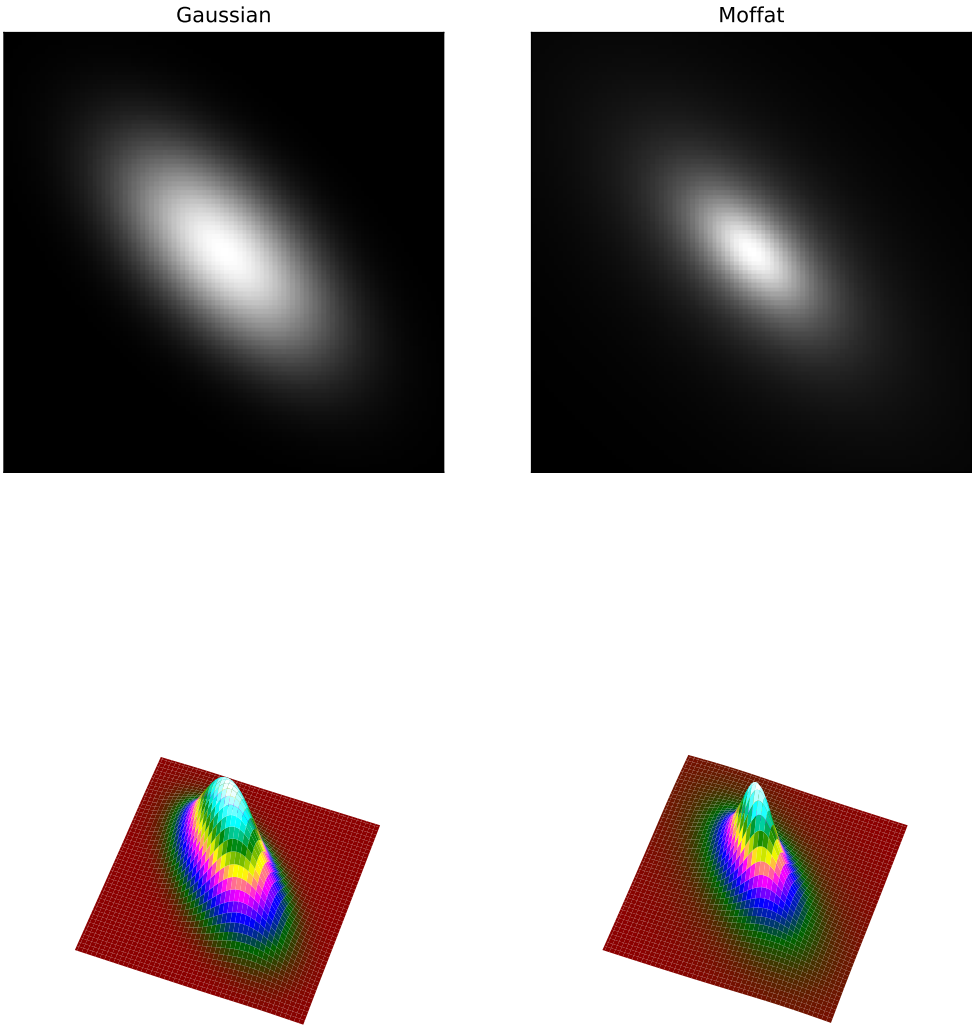


FIG. 3 – Les fonctions de rotation Gaussienne et de Moffat ont $\sigma_x = 2\sigma_y$, $\theta = 45$. Pour la fonction de Moffat, $\beta = 1.4$.

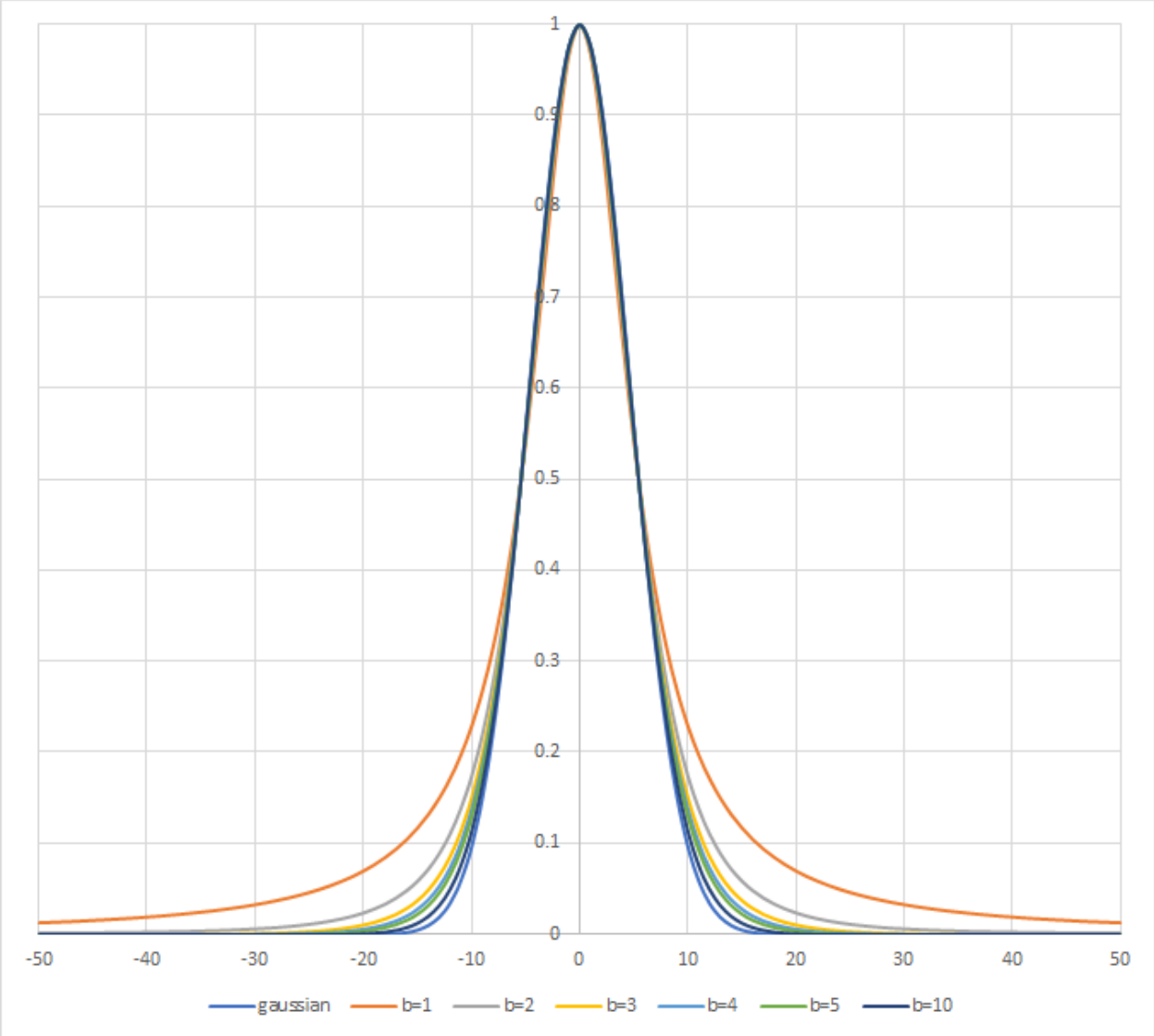


FIG. 4 – Profil des étoiles avec les modèles gaussien et de Moffat. Plusieurs valeurs de β sont essayées. $\beta = 10$ donne un profil très proche du modèle gaussien.

2. Une fonction d'ajustement PSF elliptique de Moffat définie comme suit

$$M(x,y) = B + A\left(1 + \frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right)^{-\beta}, \tag{12.2}$$

où :

— B est le fond du ciel local moyen.
— A est l'amplitude, c'est-à-dire la valeur maximale de la PSF ajustée.
— x_0, y_0 sont les coordonnées du centroïde en pixel.
— σ_x, σ_y sont la déviation standard de la distribution Gaussienne sur l'axe horizontale et verticale, mesurées en pixels.
— β est l'exposant de la formule de Moffat qui contrôle la forme globale de la fonction d'ajustement. La limite supérieure de ce paramètre a été fixée à 10. Une valeur supérieure n'a pas de sens et signifie que la gaussienne est suffisamment bonne pour s'adapter à l'étoile.

D'autres paramètres sont dérivés de ces variables ajustées :

— FWHM$_x$ et FWHM$_y$: Le Full Width Half Maximum sur l'axe X et Y en pixels. Ces paramètres sont calculés comme suit :
  — FWHM$_x = 2\sigma_x\sqrt{2\log 2}$.
  — FWHM$_y = 2\sigma_y\sqrt{2\log 2}$.
  — Il est possible d'obtenir les paramètres FWHM en secondes d'arc. Pour cela, vous devez remplir tous les champs correspondant à votre caméra et à la focale de votre lentille/télescope dans la fenêtre de réglage des paramètres du menu, puis *Informations de l'image* et *Information*. Si les mots-clés standards FITS `FOCALLEN`, `XPIXSZ`, `YPIXSZ`, `XBINNING` et `YBINNING` sont lus dans l'en-tête FITS, le PSF calculera également l'échelle de l'image en secondes d'arc par pixel.
— r : Le paramètre de rondeur. Il est exprimé par FWHM$_x$/FWHM$_y$, avec FWHM$_x$ $>$ FWHM$_y$ comme condition de symétrie.
— Un autre paramètre est également ajusté dans les modèles gaussien et de Moffat. Il s'agit de l'angle de rotation θ, défini dans l'intervalle $[-90, +90]$. L'ajout de ce paramètre implique un changement de coordonnées où les variables x et y exprimées dans (12.1) et (12.2) sont remplacées par x' et y' :

$$\begin{aligned} x' &= +x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta. \end{aligned} \tag{12.3}$$

12.3 Minimisation

La minimisation est effectuée avec un algorithme non linéaire Levenberg-Marquardt grâce à la très robuste GNU Scientific Library. Cet algorithme est utilisé pour trouver le minimum d'une fonction qui fait correspondre un ensemble de paramètres à un ensemble de valeurs observées. C'est une combinaison de deux techniques d'optimisation : la méthode de descente du gradient et la méthode inverse-hessienne.

L'algorithme de Levenberg-Marquardt ajuste le compromis entre ces deux méthodes en fonction de la courbure de la fonction à minimiser. Lorsque la courbure est petite, l'algorithme utilise la méthode de descente du gradient, et lorsque la courbure est grande, l'algorithme utilise la méthode inverse-hessienne.

Depuis la version 1.2.0, la partie saturée de l'étoile est retirée du processus d'ajustement, permettant de capturer beaucoup plus précisément la partie non saturée. C'est ce qui permet de "reconstruire" le profil de l'étoile lorsqu'on utilise l'option de menu *Désaturer* ou la commande *unclipstars*.

12.4 Utiliser

La PSF dynamique peut être appelée de deux manières différentes, selon ce que vous voulez :

Il se peut que vous ne souhaitiez ajuster qu'une ou quelques étoiles. Dans ce cas, après avoir dessiné une sélection autour d'une étoile non saturée (ceci est important pour la précision du résultat), vous pouvez soit faire un clic droit et choisir l'élément *Pointer l'étoile*, soit cliquer sur le bouton + dans le dialogue Dynamic PSF, soit taper `Ctrl` + `Espace`. Une ellipse est alors dessinée autour de l'étoile. Pour ouvrir la boîte de dialogue, il est également possible d'utiliser le raccourci `Ctrl` + `F6`.

Vous pouvez analyser autant d'étoiles que possible en cliquant sur l'icône , ou en utilisant la ligne de commande *findstar*. Toutes les étoiles détectées sont entourées d'une ellipse : orange si l'étoile est correcte, magenta si l'étoile est saturée. Il est également possible de montrer la moyenne des paramètres calculés comme illustré ci-dessous, en cliquant sur le bouton .

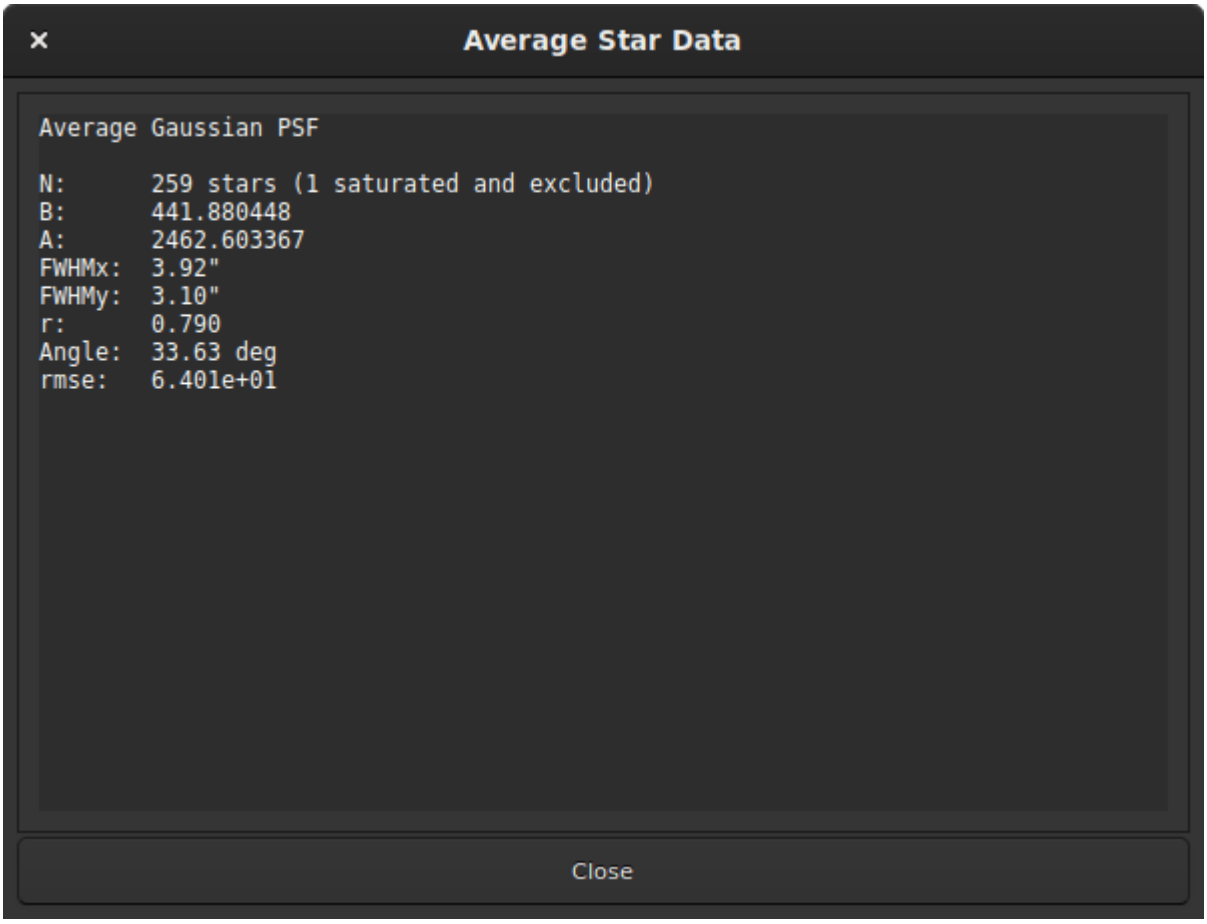


Fig. 5 – Moyenne des étoiles ajustées dans le modèle gaussien.

La détection des étoiles a un certain nombre d'utilisations :

- Siril l'utilise en interne à des fins astrométriques lors de l'alignement de la séquence d'images. Cette opération est automatique et ne nécessite aucune intervention de l'utilisateur.
- Les étoiles étant très lumineuses par rapport à des objets de faible intensité tels que les nébuleuses ou les galaxies, il est très fréquent que certaines étoiles d'une image soient saturées, ce qui signifie que leur profil de luminosité est écrêté. Cela peut poser des problèmes pour certaines fonctions de traitement d'image, en

particulier la déconvolution, et entraîne une perte d'informations chromatiques et un gonflement légèrement plus important des étoiles lors de l'application des étirements. L'analyse de toutes les étoiles vous montrera lesquelles sont saturées, et vous pourrez alors utiliser l'option de menu *Désaturer* ou la commande *unclipstars* pour résoudre le problème par synthèse de la partie écrêtée du profil.

Ligne de commande Siril

```
unclipstars
```

Reprofile les étoiles coupées de l'image chargée pour les désaturer, en mettant à l'échelle la sortie de manière à ce que toutes les valeurs de pixels soient <= 1,0

— Idéalement, toutes les étoiles d'une image devraient être parfaitement rondes, mais des problèmes tels que la coma, l'astigmatisme et un mauvais suivi, ainsi que des problèmes tels qu'une back focus incorrect des aplanisseurs de champ, peuvent donner lieu à des modèles d'étoiles elliptiques dans une image. Les ellipses produites par l'outil Dynamic PSF peuvent donner une bonne illustration visuelle de ces problèmes.
— L'examen des paramètres moyens de l'étoile, en particulier la FWHM et le paramètre bêta de la fonction d'ajustement de Moffat, peut fournir des informations sur la qualité du seeing dans une image.
— La détection de toutes les étoiles est la première étape de l'utilisation de l'outil *Traitement des étoiles → Resynthèse intégrale*. Celui-ci synthétise des profils de luminosité corrigés pour toutes les étoiles détectées, et peut être utilisé pour créer un masque d'étoiles synthétique qui peut ensuite être mélangé avec une image sans étoiles générée par Starnet++. Dans ce cas, la détection des étoiles à l'aide du profil Moffat peut donner un résultat plus réaliste et peut également faciliter le filtrage des galaxies incorrectement détectées comme des étoiles en utilisant le paramètre Minimum beta.
— Le bouton Actif/inactif **Étoile sélectionnée au centre** peut être utilisé pour trouver rapidement et facilement une étoile particulière de la liste dans l'image, en la centrant dans la fenêtre d'affichage. Ceci est utile si vous avez détecté toutes les étoiles et que vous souhaitez vérifier des solutions spécifiques pour vous assurer qu'il s'agit bien d'une étoile et non d'une galaxie ou d'un rayon cosmique.
— De même, un clic sur une ellipse d'étoile orange ou magenta dans la fenêtre principale mettra en surbrillance l'étoile dans la boîte de dialogue Dynamic PSF. Cela peut être utile si vous souhaitez voir les paramètres d'une étoile individuelle.
— Les fonctions de déconvolution de Siril permettent d'utiliser les mesures de l'outil PSF dynamique pour synthétiser une fonction de déconvolution qui correspond aux paramètres des étoiles mesurés directement à partir de l'image.

12.5 Configuration

Le PSF dynamique peut être configuré à l'aide des paramètres de la boîte de dialogue Dynamic PSF :

— **Rayon** définit la demi-taille de la boîte de recherche. Si vous avez des difficultés à détecter des étoiles particulières, vous pouvez essayer de modifier cette valeur, mais la valeur par défaut convient généralement.
— **Seuil** modifie le seuil au-dessus du bruit pour la détection des étoiles. Si vous augmentez cette valeur, moins d'étoiles faibles seront détectées. Cependant, vous pouvez toujours le faire pour les images très bruitées. Diminuer cette valeur peut détecter plus d'étoiles faibles, mais rendra l'algorithme plus susceptible de mal identifier les pics de bruit aléatoires comme des étoiles.
— **Rondeur** définit l'ellipticité admissible pour que les détections soient acceptées comme des étoiles. Les étoiles très elliptiques peuvent être dues à un suivi imparfait ou à des aberrations, mais il arrive aussi que des étoiles trop proches les unes des autres soient détectées comme une seule étoile très allongée. Pour mettre en évidence tous ces problèmes, il est possible d'activer une limite supérieure pour la rondeur. Une valeur maximale de 1 équivaut à désactiver la plage, ne laissant que la valeur minimale. Cette plage de rondeur doit être désactivée pour l'alignement ou l'astrométrie.
— **Convergence** définit un critère utilisé par le solveur. L'augmenter permet au solveur d'avoir plus d'interations pour converger et peut potentiellement détecter des étoiles supplémentaires, mais peut augmenter son temps

d'exécution.
— Le type de profil permet de choisir entre la résolution de profils **Gaussienne** ou **Moffat** pour les étoiles.
— **Beta minimum** définit une valeur minimale admissible de bêta pour qu'une détection soit acceptée comme une étoile. Les galaxies peuvent parfois être détectées comme des étoiles de profil Moffat, mais elles ont des profils diffus et la valeur de beta est généralement très faible, inférieure à environ 1,5.
— **Assouplir les contrôles PSF** permet de relâcher plusieurs des vérifications de qualité des étoiles candidates. Cela est susceptible d'entraîner une augmentation significative des détections d'étoiles faussement positives, souvent avec des paramètres aberrants.
— Une gamme d'amplitude **minimum** et **maximum** peut être définie pour limiter l'amplitude (paramètre nommé "A" dans les rapports) des étoiles détectées. Cela est utile si seules les étoiles non saturées doivent être sélectionnées, par exemple pour l'ajustement de PSF dans la déconvolution. Notez que le retrait des étoiles saturées de la détection peut perturber l'alignement et l'astrométrie.

Astuce

Les paramètres définis dans cette fenêtre peuvent être testés sur l'image actuellement chargée. Cependant, vous devez garder à l'esprit qu'ils seront également utilisés pour toutes les images de la séquence, notamment pour de l'alignement global.

La commande *findstar* obéit aux mêmes paramètres entrés dans la boîte de dialogue PSF dynamique, mais elle peut également être configurée à l'aide de la commande *setfindstar*.

Ligne de commande Siril

```
findstar [-out=] [-layer=] [-maxstars=]
```

Détecte les étoiles dans l'image actuellement chargée, dont le niveau est supérieur à un seuil calculé par Siril.
Ensuite, une PSF est appliquée et Siril rejette toutes les structures détectées qui ne répondent pas à un ensemble de critères de détection prescrits, qui peuvent être ajustés avec la commande SETFINDSTAR.
Enfin, une ellipse est dessinée autour des étoiles détectées.

Le paramètre optionnel **-out=** permet d'enregistrer les résultats dans le chemin d'accès donné.
L'option **-layer=** spécifie la couche sur laquelle la détection est effectuée (pour les images en couleur uniquement).
Vous pouvez également limiter le nombre maximum d'étoiles détectées en passant une valeur à l'option **-maxstars=**.

Voir aussi la commande CLEARSTAR

Liens : *psf*, *setfindstar*, *clearstar*

Ligne de commande Siril

```
setfindstar [reset] [-radius=] [-sigma=] [-roundness=] [-focal=] [-pixelsize=] [-
↪convergence=] [ [-gaussian] | [-moffat] ] [-minbeta=] [-relax=on|off] [-minA=] [-
↪maxA=] [-maxR=]
```

Définit le paramètre de détection d'étoiles pour les commandes FINDSTAR et REGISTER.

Si vous ne passez aucun paramètre, vous obtiendrez la liste des valeurs actuelles.
Passer **reset** réinitialise toutes les valeurs aux valeurs par défaut. Vous pouvez alors toujours passer des valeurs après ce mot-clé.

Valeurs paramétrables :

-radius= définit le rayon de la boite de recherche initiale et doit être compris entre 3 et 50.
-sigma= définit le seuil au-dessus du bruit et doit être supérieur ou égal à 0,05.
-roundness= définit la rondeur minimale des étoiles et doit être comprise entre 0 et 0,95. **-maxR** permet de définir une limite supérieure à l'arrondi, pour visualiser uniquement les zones où les étoiles sont significativement allongées, ne pas utiliser pour l'alignement.
-minA et **maxA** définit la limite du minimum et maximum de l'amplitude des étoiles à conserver, normalisé entre 0 et 1.
-focal= définit la longueur focale du télescope.
-pixelsize= définit la taille des pixels du capteur.
-gaussian et **-moffat** configure le modèle de solveur utilisé (Gaussian par défaut).
Si Moffat est sélectionné, **-minbeta=** définit la valeur minimale de beta pour laquelle les étoiles candidates seront acceptées et doit être supérieure ou égale à 0,0 et inférieure à 10,0.
-convergence= définit le nombre d'itération réalisée pour ajuster la PSF et doit être compris entre 1 et 3 (plus tolérant).
-relax= relâche les vérifications qui sont faites sur les étoiles candidats pour évaluer s'ils sont des étoiles ou non, pour permettre aux objets qui n'ont pas la forme d'une étoile d'être quand même acceptés (désactivé par défaut)

Liens : *findstar*, *register*, *psf*

12.6 Références

CHAPITRE 13

Entête FITS

Les mots-clés contenus dans l'en-tête d'un fichier FITS peuvent être affichés dans Siril. Pour ce faire, il suffit de cliquer sur *Outils* → *Entête FITS*.

Since version 1.3.0, it is possible to modify the value of keywords supported by Siril, either with the *update_key* command, or via the GUI, in the appropriate window. This window is divided into two tabs. The first, the *Header Editor*, is active for single images, out of sequence. It may vary from what the file header actually looks like, and represents more the state it will be in after saving. Keyword values are updated in real time. The second tab displays the header in text format as written in the file and as it was represented for Siril versions below 1.3.0.

Avertissement

Les mots-clés HISTORY ne sont pas affichés dans l'éditeur. Ils sont visible dans l'onglet *Entête sauvgardé*.

Une autre méthode pour afficher l'entête est d'utiliser la ligne de commande *dumpheader*. Ceci affiche le même entête que l'onglet *Entête sauvegardé*.

Avertissement

Comme les fichier SER contiennent très peu de mots-clés et sont différents des fichier FITS, cette commande n'est pas applicable sur ce type de séquence.

Les mots-clés, valeurs et commentaire peuvent être modifié, cependant, certain mots-clés peuvent être protégés. Pour les reconnaître facilement, les mot-clés protégé sont affiché en couleur saumon.

Editing is very simple, just double-click on the cell to be modified. The first time selects the field, the second opens the edit mode. Pressing the `Enter` key validates the entry. To make the changes effective, remember to save the file.

Avertissement

Veuillez noter que Siril ne vérifie pas la validité de la valeur saisie. Il appartient à l'utilisateur de saisir une valeur valide. Une valeur incorrecte peut entraîner un comportement indésirable dans le traitement des mots-clés par Siril.

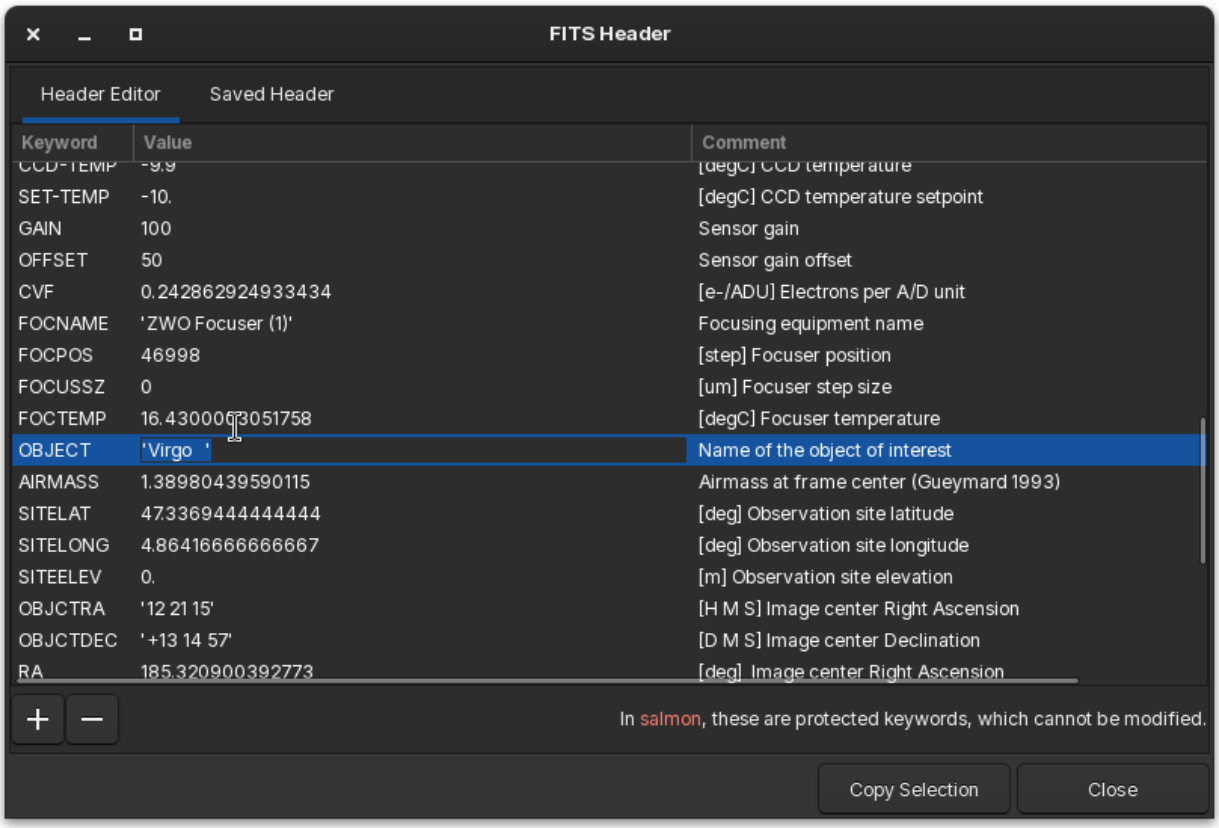


Fig. 1 – Fenêtre d'en-tête FITS lors de l'édition de la valeur d'un mot-clé.

La fenêtre contient une option sous la forme d'un bouton *Copier la sélection*. Il permet de copier les lignes sélectionnées dans le presse-papiers dans le format originale de l'entête FITS, et fonctionne pour les deux onglets.

Enfin, vous pouvez ajouter un nouveau mot-clé à l'aide du bouton + en bas à gauche , et en supprimer à l'aide du bouton - :

1. To add a keyword, click on the + button. A new dialog box opens, as shown below. This window lets you add a new keyword, whose name will be limited to 8 characters (the `HIERARCH` convention is not used). You can leave the *Comment* field empty, if the other two are filled in. However, the user can only fill in the *Comment* field to add a single comment to the FITS header. Finally, if the keyword exists, then its value will be updated with the information provided.
2. To delete a keyword, select one or more lines and click on the - button (or with the keys `Del` or `Backspace`). The selected keywords will be deleted, unless they are protected.

> **Astuce**
>
> Lorsqu'une séquence est chargée, il n'est possible de sélectionner qu'une seule ligne à la fois. Cela signifie que vous ne pouvez supprimer qu'un seul mot-clé à la fois dans une séquence. En revanche, pour une image unique, vous pouvez en sélectionner plusieurs.

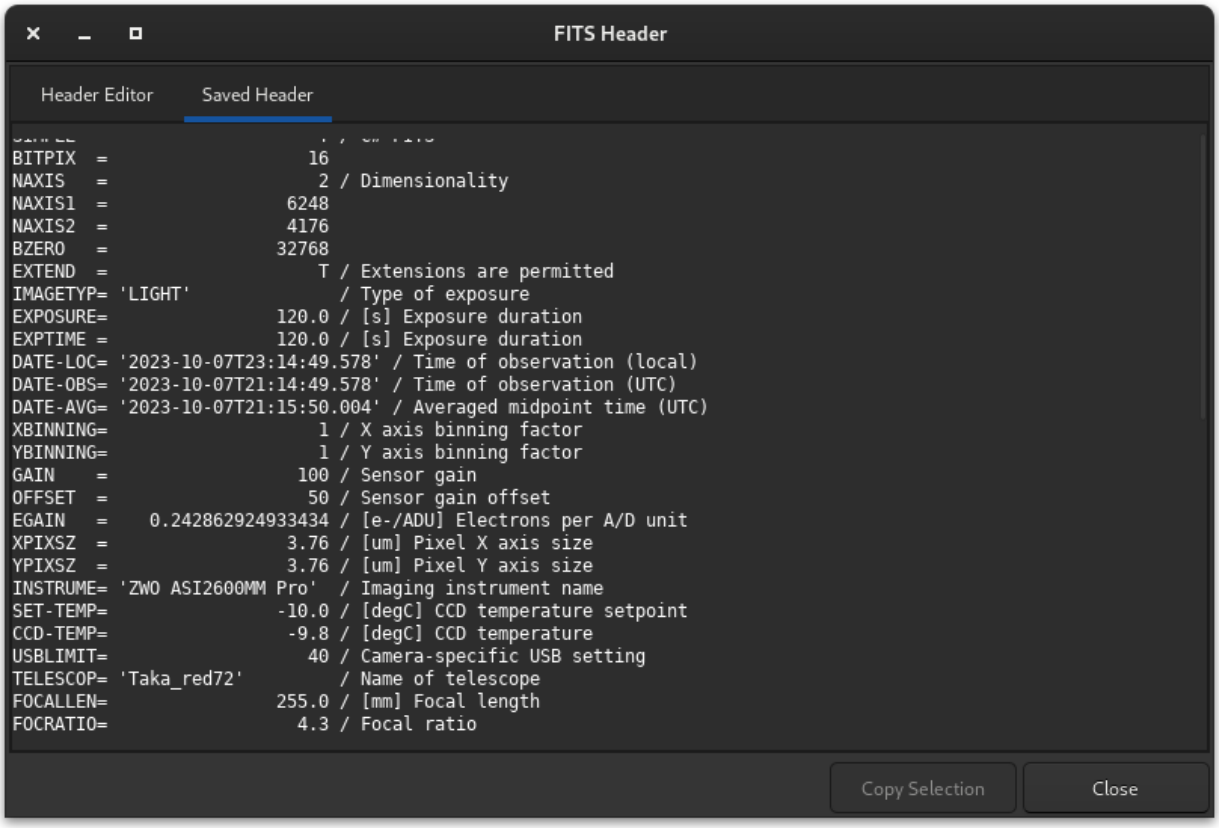


Fig. 2 – Second onglet de la boite de dialogue de l'entête FITS.

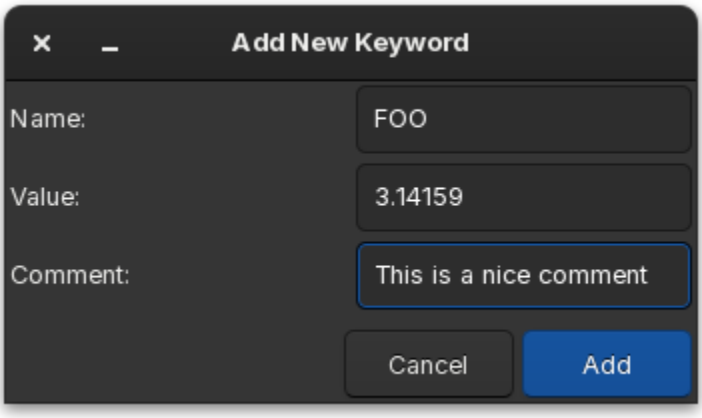


Fig. 3 – La fenêtre pour ajouter de nouveaux mot clé.

Ligne de commande Siril

```
dumpheader
```

Affiche l'en-tête FITS de l'image chargée dans la console

Ligne de commande Siril

```
update_key key value [keycomment]
update_key -delete key
update_key -modify key newkey
update_key -comment comment
```

Updates FITS keyword. Please note that the validity of **value** is not checked. This verification is the responsibility of the user. It is also possible to delete a key with the **-delete** option in front of the name of the key to be deleted, or to modify the key with the **-modify** option. The latter must be followed by the key to be modified and the new key name. Finally, the **-comment** option, followed by text, adds a comment to the FITS header. Please note that any text containing spaces must be enclosed in double quotation marks

Ligne de commande Siril

```
sequpdate_key sequencename key value [keycomment]
sequpdate_key sequencename -delete key
sequpdate_key sequencename -modify key newkey
sequpdate_key sequencename -comment comment
```

Same command as UPDATE_KEY but for the sequence **sequencename**. However, this command won't work on SER sequence

Links : *update_key*

CHAPITRE 14

Profil d'intensité

Siril has an intensity profiling mode. The user selects a line between two points and Siril will generate a graph of the pixel values between them. This has several uses. It can be used to inspect the intensity profile of an individual star or a whole galaxy.

14.1 Profil d'intensité de base

Pour établir un profil d'intensité de base d'une étoile ou d'un autre objet, sélectionnez le bouton Profil dans la barre d'outils inférieure. Siril passe alors en mode profilage et ouvre une petite boîte de dialogue.

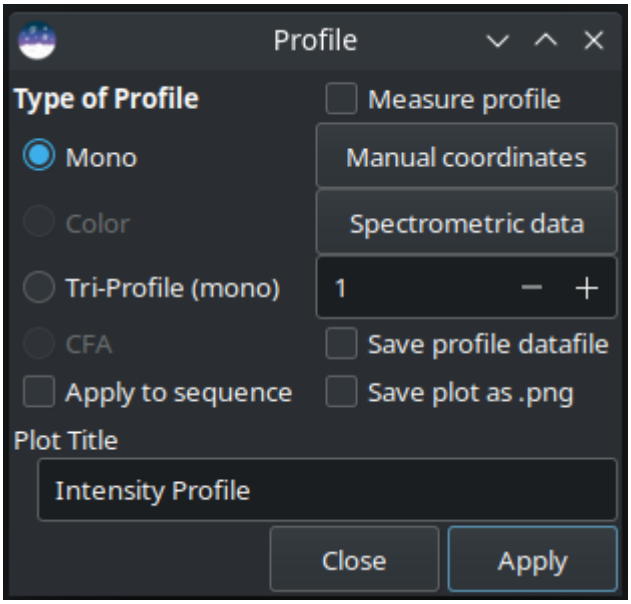
Vous pouvez maintenant cliquer sur l'image principale et la faire glisser pour définir les points de départ et d'arrivée de la ligne que vous souhaitez profiler. Si vous maintenez la touche `Maj` enfoncée pendant que vous faites glisser la ligne, celle-ci se déplacera horizontalement ou verticalement.

Astuce

Lorsque la ligne de profil est exactement horizontale ou exactement verticale, les valeurs exactes des pixels peuvent être utilisées directement à partir de l'image. Lorsque la ligne de profil n'est ni horizontale ni verticale, les points à tracer ne tombent pas exactement sur un pixel et des valeurs de pixels interpolées de manière bilinéaire sont donc utilisées.

Un titre personnalisé pour votre graphique peut être saisi dans le contrôle au bas de la boîte de dialogue.

Astuce



Lors du traitement d'une séquence, il est possible d'afficher le numéro de l'image et le total dans le titre personnalisé en ajoutant () à la fin du titre. Par exemple, en saisissant **Spectre solaire ()** comme titre pour une séquence de 5 images, les titres **Spectre solaire (1 / 5)**, **Spectre solaire (2 / 5)**, etc. seront générés. Les parenthèses sont ignorées et supprimées lors du traitement d'une seule image.

14.2 Types de profils

Utilisez les cases d'option pour sélectionner le type de profil que vous souhaitez. (Cliquez sur les exemples d'images ci-dessous pour les voir en taille réelle.)

— **Profil mono**. Pour les images mono ou couleur, générer un profil de luminance entre deux points. Ce mode peut être utilisé avec des données spectrométriques.

Astuce

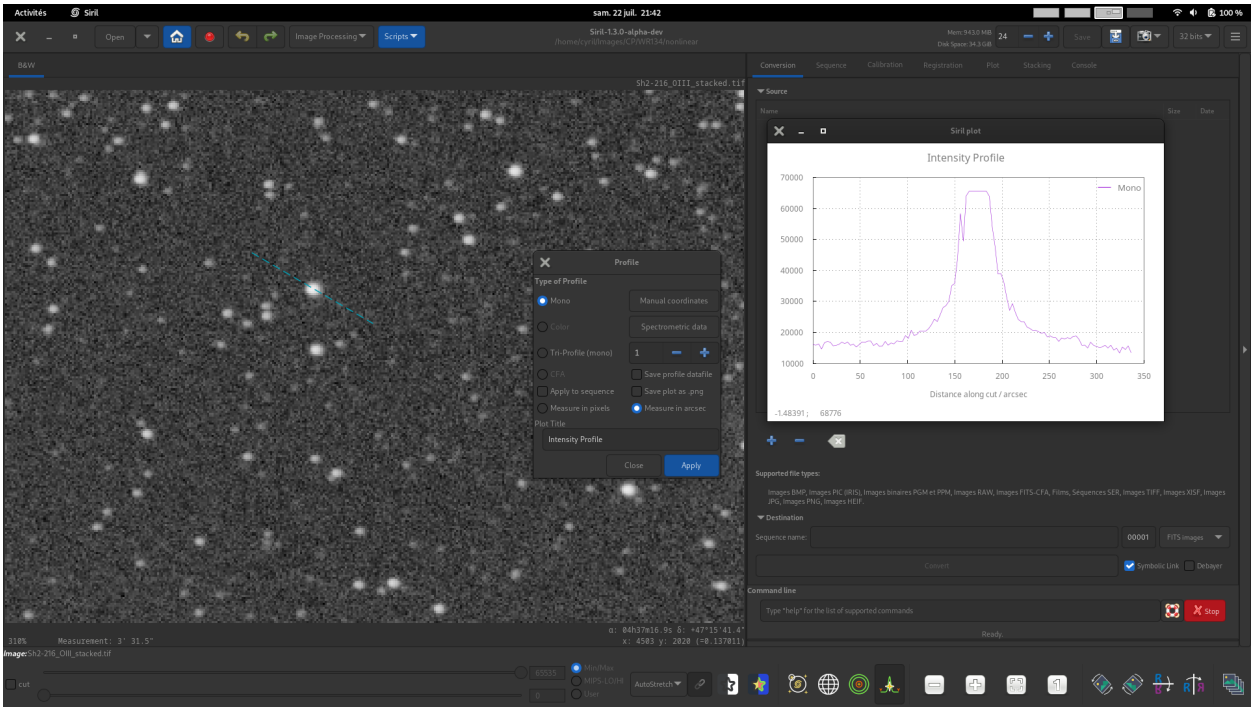
Si une image couleur est chargée mais que le mode de profilage mono est sélectionné, le profil sera établi en fonction de la fenêtre d'affichage. Les fenêtres R, G et B fournissent des profils mono de leur canal respectif et la fenêtre RVB fournit un profil de luminance pondérant les 3 canaux de manière égale.

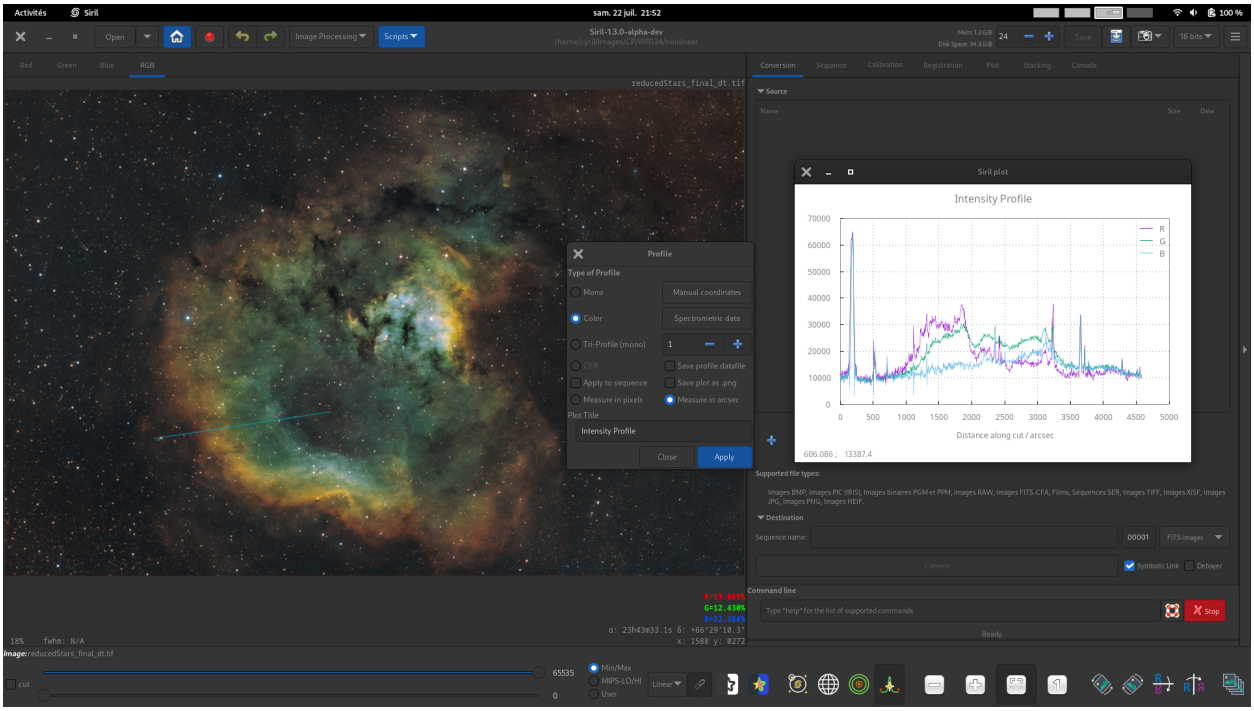
— **Profil de couleur**. Pour les images couleur, générer trois profils pour les valeurs des pixels R, V et B entre deux points. Ce mode peut être utilisé avec des données spectrométriques.

— **Tri-profil (mono)**. Pour les images mono ou couleur, génère trois profils de luminance parallèles et équidistants entre deux points. L'espacement entre les 3 profils peut être réglé à l'aide du bouton adéquate.

— **CFA**. Pour les images comportant uniquement un motif de Bayer, générer quatre profils pour les quatre sous-canaux CFA entre deux points. Cela peut s'avérer particulièrement utile pour inspecter le profil des flats à motif de Bayer ou d'autres images à motif de Bayer avant qu'elles ne soient débayerisées.

Cliquer sur *Appliquer* pour générer le profil.


Activités
Siril
sam. 22 juil. 21:42
Image Processing
Scripts
B&W
Sh2-216_OIII_stacked.tif
Profile
Type of Profile
Mono
Manual coordinates
Spectrometric data
Tri-Profile (mono)
Save profile datafile
Apply to sequence
Save plot as .png
Measure in pixels
Measure in arcsec
Plot Title
Intensity Profile
Close
Apply
Siril plot
Intensity Profile
Mono
Distance along cut / arcsec
Conversion
Sequence
Calibration
Registration
Plot
Stacking
Console
Source
Destination
Command line
Ready.


Activités
Siril
sam. 22 juil. 21:52
Image Processing
Scripts
Red
Green
Blue
RGB
reducedStars_final_dt.tif
Profile
Type of Profile
Mono
Manual coordinates
Color
Spectrometric data
Tri-Profile (mono)
Save profile datafile
Apply to sequence
Save plot as .png
Measure in pixels
Measure in arcsec
Plot Title
Intensity Profile
Close
Apply
Siril plot
Intensity Profile
R
G
B
Distance along cut / arcsec
Conversion
Sequence
Calibration
Registration
Plot
Stacking
Console
Source
Destination
Command line
Ready.

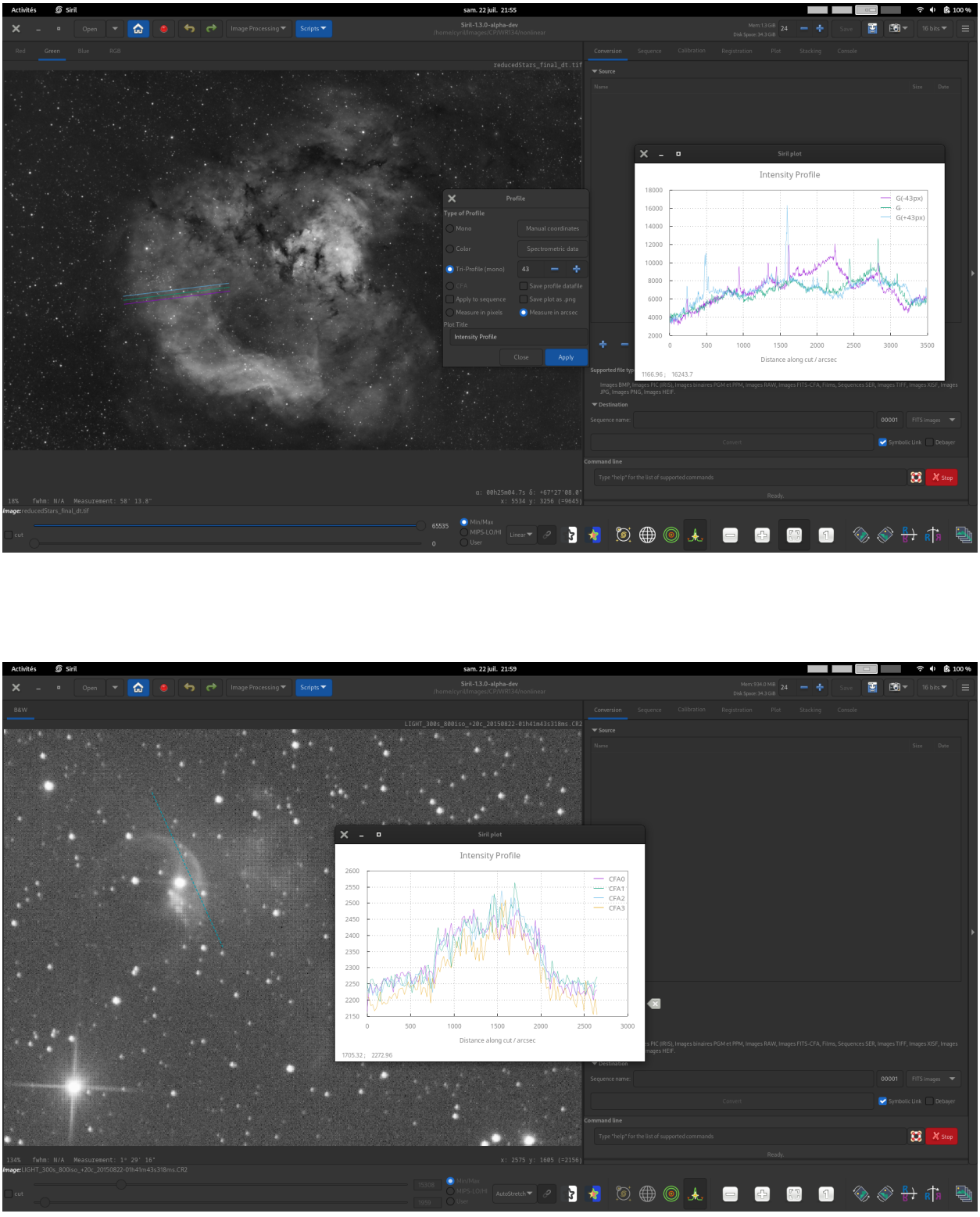


FIG. 1 – Cette image illustre l'utilisation du contrôle Titre personnalisé pour définir un titre personnalisé pour le graphe.

14.3 Saisie de coordonnées précises

Afin de faciliter la saisie des coordonnées de manière précise et répétée, une méthode de saisie manuelle est proposée. Cliquez sur le bouton *Coordonnées manuelles* et vous pourrez entrer les coordonnées X et Y des points de départ et d'arrivée de la ligne de profil. Si une ligne de profil est déjà tracée mais qu'un point n'est pas tout à fait à l'endroit voulu, vous pouvez utiliser cette boîte de dialogue pour affiner l'emplacement des points d'extrémité.

If you wish to set an endpoint exactly to the position of a star, make a rectangular selection around the star and click the relevant star button to the right of the dialog.

> **Note**
>
> Lorsque l'on utilise le mode CFA, les coordonnées sont données dans l'image *d'entrée*. Cependant, chaque canal CFA représente la moitié de la largeur et de la hauteur. L'axe des x dans le tracé du mode CFA est mesuré en pixels dans le sous-canal CFA, c'est-à-dire qu'il couvre la moitié du nombre de pixels qu'il couvre dans l'image d'entrée.

14.4 Mesures

La ligne de profil d'intensité peut être utilisée comme outil de mesure de deux manières :

- En cochant la case *Mesurer le profil*, toutes les lignes de profil créées avec la souris seront mesurées, de manière similaire à la fonction de mesure rapide `Ctrl-Maj-Glisser`.
- Dans la boîte de dialogue Coordonnées, il y a un bouton *Mesurer*. Il fournit la même fonction de mesure, mais vous permet de définir exactement les points d'extrémité, puis de mesurer la ligne de profil à la demande. En sélectionnant des étoiles, des planètes mineures ou des noyaux de comète comme points d'extrémité, comme décrit ci-dessus, les mesures entre deux corps célestes peuvent être effectuées très précisément (avec une précision inférieure au pixel).

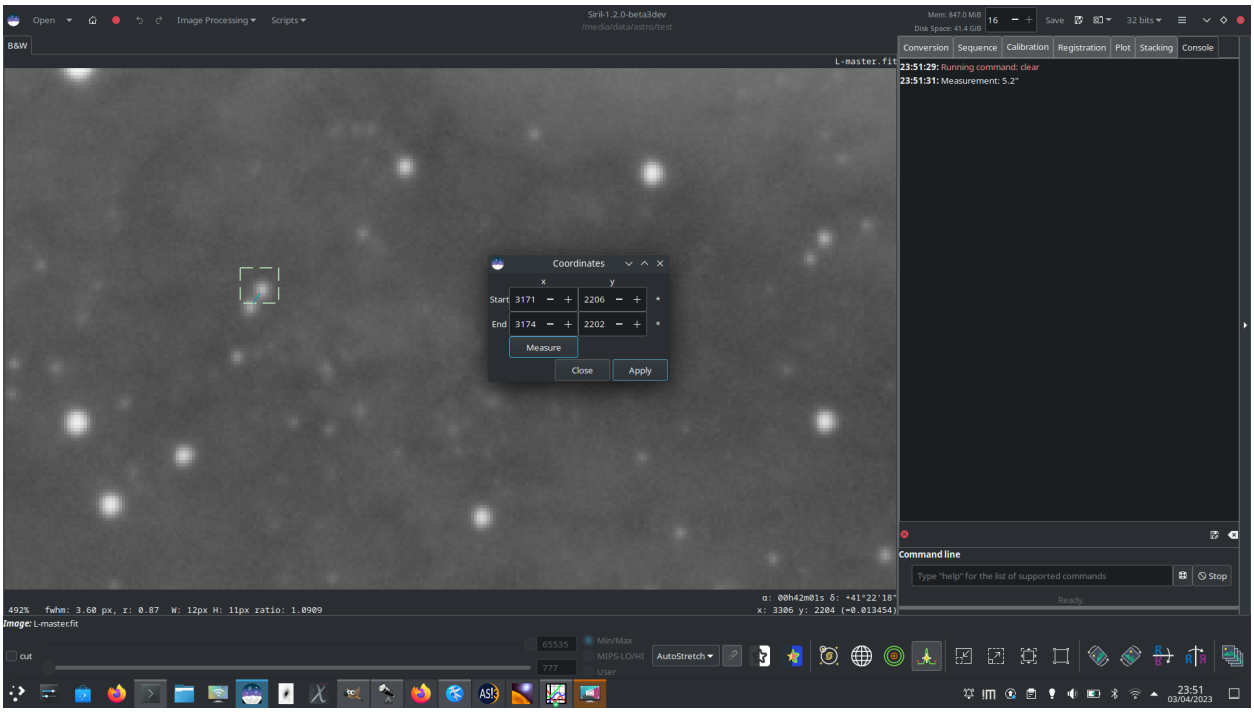


Fig. 2 – Ici, deux étoiles proches ont été sélectionnées et placées comme points d'extrémité, et la séparation entre elles a été mesurée à 5,2 arcsec. Cette méthode pourrait être utilisée pour étudier les binaires proches ou pour trianguler la position d'une planète mineure.

Note

La fonction de mesure de Siril fait l'approximation du petit angle pour la séparation angulaire θ. Le terme d'erreur le plus significatif est proportionnel à θ^3 et est inférieur à 1% pour les mesures jusqu'à 10° : il est donc valide pour la plupart des utilisations astrométriques, mais deviendra imprécis pour les grandes mesures sur des images à très grand champ. Un avertissement sera écrit dans les logs pour les mesures supérieures à 10°.

14.5 Outil graphique de Siril

La fonction de profilage utilise l'outil graphique interne de Siril pour afficher les différents profils. Avec les fichiers `*.dat` produits, vous pouvez toujours utiliser l'outil graphique de votre choix pour explorer les données sous-jacentes.

Un `clic droit` n'importe où dans la surface du graphique fera apparaître un menu contextuel pour :

- Afficher/masquer la grille et la légende
- Exporter la vue actuelle dans le presse-papiers, `*.png` ou `*.svg`
- Sauvegarde des données sous-jacentes dans un fichier `*.dat`

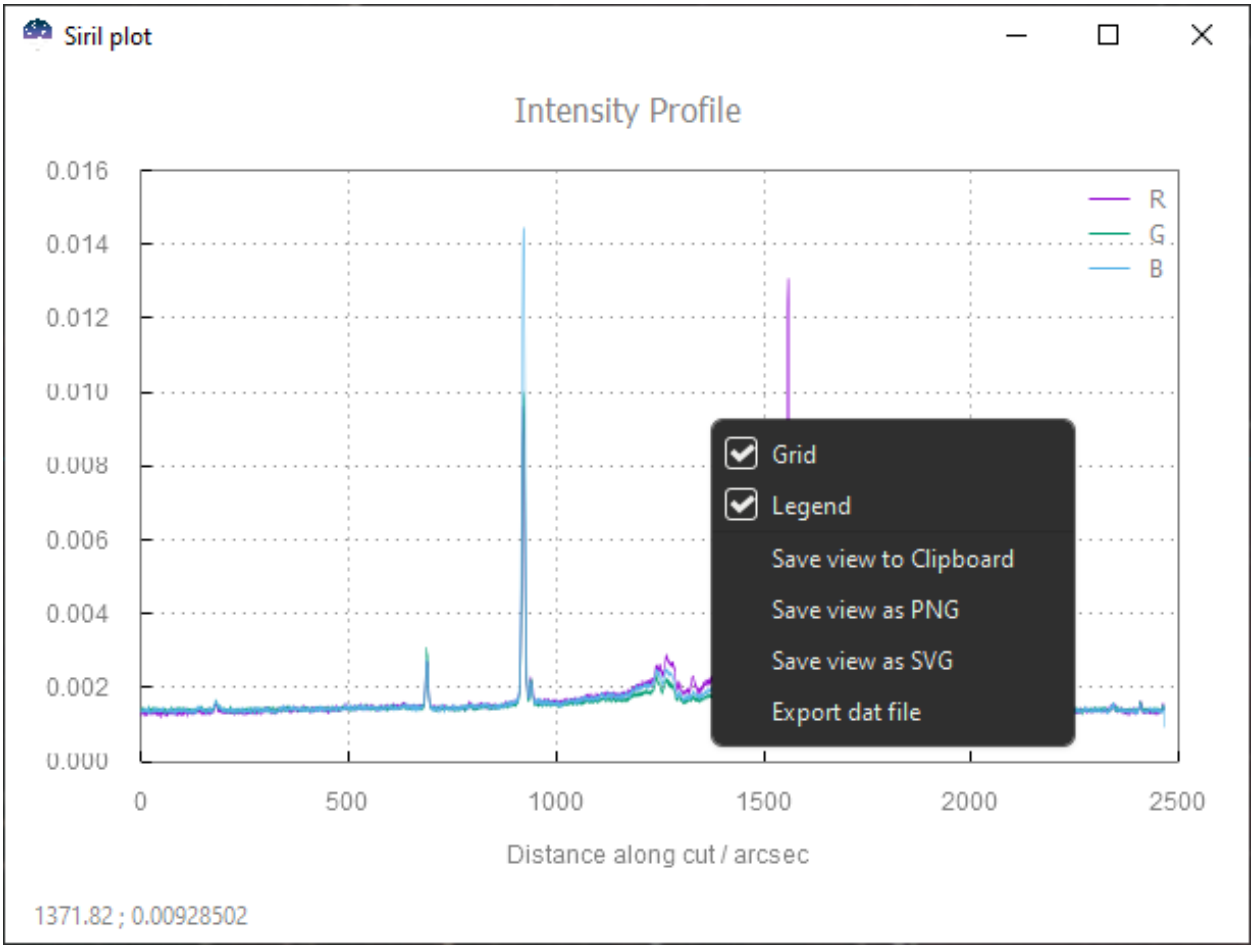


FIG. 3 – Menu contextuel du graphe de Siril

Notez que toutes les exportations tiennent compte du zoom/pan actuel, tandis que l'enregistrement dans un fichier dat exporte des données non filtrées.

Les interactions suivantes de l'interface graphique sont disponibles :

— `Cliquer` + `glisser` pour dessiner une sélection. Le zoom est réglé sur la zone sélectionnée lorsque la souris est relâchée.
— `Cliquer` + `glisser` pour effectuer un panoramique de la vue courante.
— `Ctrl` + `Défilement` pour zoomer/dézoomer.
— `Double-clic` pour rétablir la position/le zoom par défaut.

14.6 Commandes

Ligne de commande Siril

```
profile -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] { [-savedat] | [-filename=] } [-
↪layer=] [-width=] [-spacing=] ["-title=My Plot"]
```

Génère un tracé du profil d'intensité entre 2 points de l'image, également connu sous le nom de *coupure*. Les arguments peuvent être fournis dans n'importe quel ordre. Les arguments **-to=x,y** et **-from=x,y** sont obligatoires.

The argument **-layer={red | green | blue | lum | col}** specifies which channel (or luminance or colour) to plot if the image is color. It may also be used with the **-tri** option, which generates 3 parallel equispaced profiles each separated by **-spacing=** pixels, but note that for tri profiles the **col** option will be treated the same as **lum**.

L'option **-cfa** sélectionne le mode CFA, qui génère 4 profils : 1 pour chaque canal CFA dans une image avec motif de Bayer. Cette option ne peut pas être utilisée avec des images couleur ou des images mono sans motif de Bayer, et ne peut pas être utilisée en même temps que l'option **-tri**.

The option **-arcsec** causes the x axis to display distance in arcsec, if the necessary metadata is available. If not provided or if metadata is not available, distance will be shown in pixel units.

L'argument **-savedat** permet de sauvegarder les fichiers de données : le nom du fichier sera écrit dans le log. L'argument **-filename=** peut également être utilisé pour spécifier un nom de fichier dans lequel écrire le fichier de données. (L'option **-filename=** implique **-savedat**.)

L'argument **"-title=Mon Titre "** définit un titre personnalisé "Mon Titre"

Ligne de commande Siril

```
seqprofile sequence -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] [-savedat] [-layer=] [-
↪width=] [-spacing=] [ {-xaxis=wavelength | -xaxis=wavenumber } ] [{-wavenumber1= | -
↪wavelength1=} -wn1at=x,y {-wavenumber2= | -wavelength2=} -wn2at=x,y] ["-title=My␣
↪Plot"]
```

Génère un profil d'intensité entre 2 points de chaque image de la séquence. Après le premier argument obligatoire indiquant la séquence à traiter, les autres arguments sont les mêmes que pour la commande **profile**. Si l'on traite une séquence et que l'on souhaite que le numéro de l'image courante et le nombre total d'images soient affichés dans le format "Ma séquence (1 / 5)", le titre donné doit se terminer par () (par exemple, "Ma séquence ()" et les nombres seront remplis automatiquement.)

CHAPITRE 15

Inspecteurs d'images

Siril dispose de plusieurs outils qui peuvent vous aider à analyser votre image et vous renseigner sur la qualité de la prise de vue. Notamment si votre setup présente ou non des défauts optiques.

15.1 Tilt

Le premier outil proposé par Siril est le calcul du tilt. Le tilt du capteur se produit lorsque le capteur n'est pas orthogonal au plan d'imagerie : cela nécessite une intervention sur le système optique. Vous pouvez exécuter cette fonctionnalité de deux manières différentes. Soit via l'interface graphique (dans le menu *Outils* → *Analyse de l'image* → *Montrer le tilt*), soit via la ligne de commande. Cette dernière offre même une alternative qui permet de calculer le tilt sur toute une séquence d'images pour plus de précision. La commande suivante :

Ligne de commande Siril

```
tilt [clear]
```

Calcule le tilt du capteur comme la différence FWHM entre les valeurs moyennes tronquées des meilleurs et des pires coins. L'option **clear** permet d'effacer le dessin

produira un résultat :

```
22:28:13: Running command: tilt
22:28:13: Findstar: processing for channel 0...
22:28:15: Stars: 7598, Truncated mean[FWHM]: 3.40, Sensor tilt[FWHM]: 0.31 (9%), Off-
↪axis aberration[FWHM]: 0.39
```

Dans la console est indiqué :

- le nombre d'étoiles utilisées pour les mesures
- la FWHM moyenne du capteur, exempt des valeurs aberrantes

— le tilt, exprimé comme la différence entre la meilleur et la pire FWHM dans les 4 coins de l'image avec entre parenthèse le pourcentage de déviation du tilt (une valeur supérieur à 10% indique un problème de tilt)
— l'aberration, exprimé par la différence de FWHM entre les étoiles aux centre et les étoiles sur les bords du capteur

Si le nombre d'étoiles détectées est faible (<200), les paramètres de détection dynamique de la PSF permettent une amélioration en ajustant le seuil / rayon. En effet, plus le nombre d'étoiles utilisées dans le calcul est important, plus le résultat de l'analyse est fiable.

Avertissement

Pour que le résultat ait un sens, il est préférable d'exécuter cette commande sur une image et non un résultat d'empilement. Une image prétraitée (juste dématricée pour les capteurs de couleurs) est donc idéale. De plus, le quadrilatère dessiné a ses proportions exagérées, afin d'être plus visible à l'écran. Il ne peut pas correspondre exactement à la réalité.

Fig. 1 – Affichage du diagramme du tilt

Astuce

En plus de la commande *tilt -clear*, le diagramme de tilt peut être effacé en utilisant le bouton *Supprimer* dans la boîte de dialogue PSF dynamique.

Ligne de commande Siril

```
seqtilt sequencename
```

Même commande que TILT mais pour la séquence **sequencename**. Elle donne généralement de meilleurs résultats

Liens : *tilt*

15.2 Inspecteur d'aberration

Cet outil crée une mosaïque 3x3 du centre, des coins et des bords de l'image. Cela permet de comparer facilement la forme de l'étoile dans différentes sections de l'image. Vous pouvez accéder à cette fonction avec le menu *Outils* → *Analyse de l'image*→ *Inspecteur d'aberration*. Vous pouvez changer les paramètres de cet outil, pour modifier la taille des panneaux et de la fenêtre, dans les *préférences*.

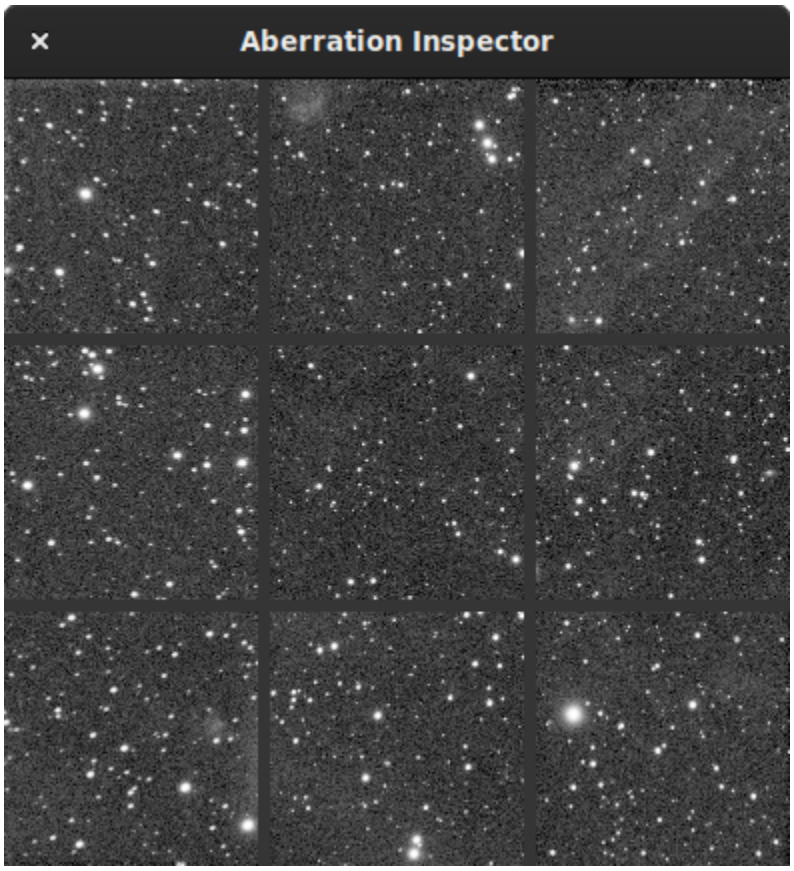


Fig. 2 – Fenêtre d'inspection d'aberration montrant l'aberration des étoiles situées sur les bords à cause du système optique.

C'est également un très bon indicateur pour savoir si l'image contient un gradient : les différences de luminosité devenant très visibles.

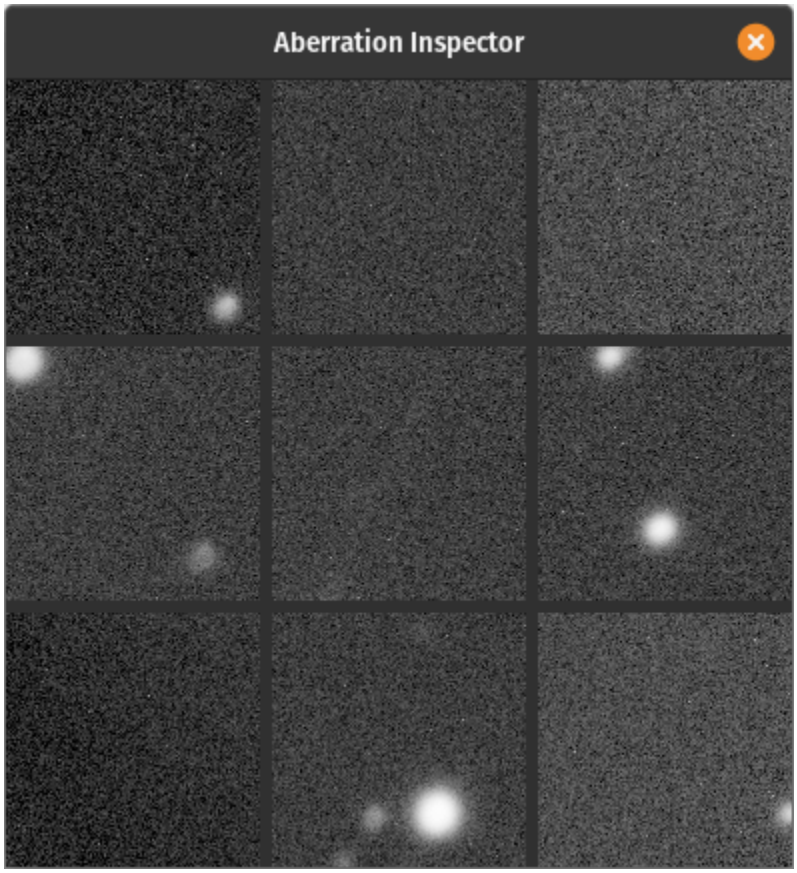


FIG. 3 – Fenêtre d'inspection d'aberration montrant une différences de luminosité.

Ligne de commande Siril

```
inspector
```

Divise l'image chargée en une mosaïque de neuf panneaux montrant les coins et le centre de l'image pour une inspection plus approfondie (interface graphique uniquement)

CHAPITRE 16

Photométrie

Cette section vous présente tous les utilitaires liés à la photométrie, en expliquant d'abord les principes de la photométrie, puis comment elle est utilisée dans Siril.

Siril est capable de déterminer la magnitude des étoiles ainsi que son incertitude. A partir de là, il est possible d'étudier la variabilité de certaines étoiles, des exoplanètes ou des occultations. Une courbe de lumière est également construite à la fin du processus.

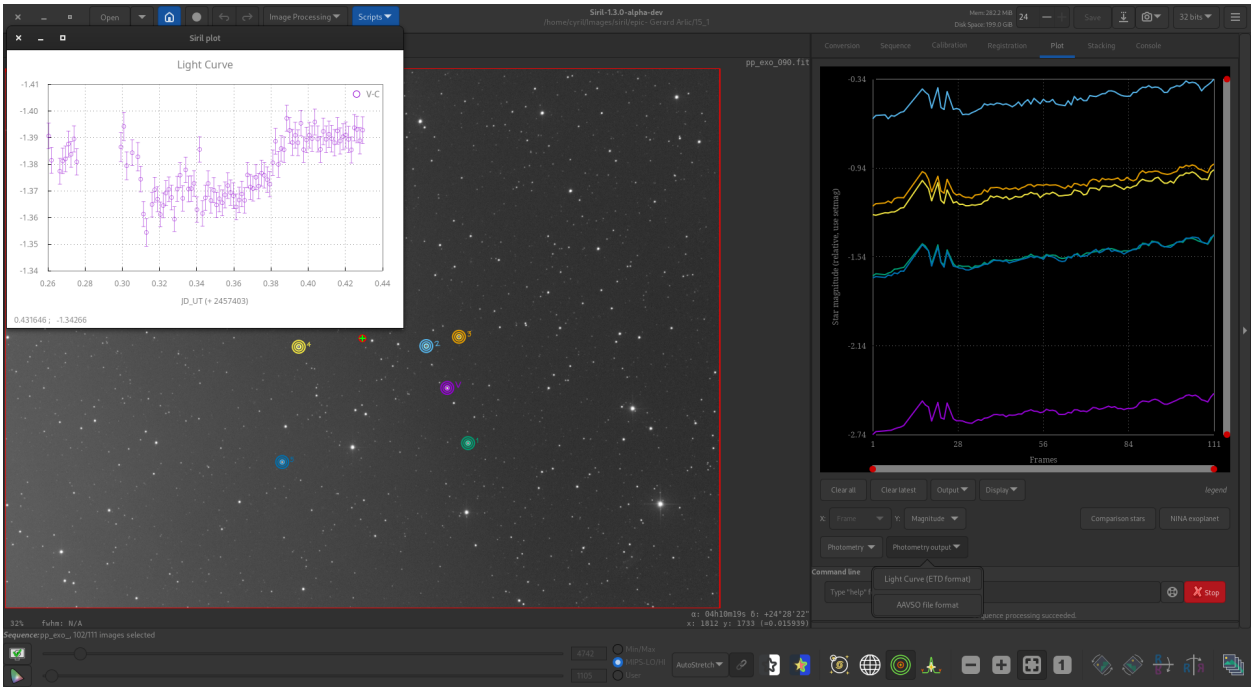


FIG. 1 – Exemple de photométrie d'exoplanète dans Siril.

16.1 Principes

La photométrie est la science de la mesure de la lumière. Elle vise à mesurer le flux ou l'intensité de la lumière émise par les objets astronomiques. Dans Siril, la photométrie peut être utilisée pour analyser la courbe de lumière des étoiles variables, les transits d'exoplanètes ou les occultations d'étoiles, ou pour calibrer les couleurs dans les images RVB.

La photométrie d'ouverture est la méthode utilisée. Son principe de base consiste à additionner le flux observé dans un rayon donné à partir du centre d'un objet, puis à soustraire la contribution totale du fond du ciel dans la même région (calculée dans l'anneau entre les rayons intérieur et extérieur, en excluant les pixels déviants), ce qui ne laisse que le flux de l'objet pour calculer une magnitude instrumentale. Ceci est illustré dans la figure suivante.

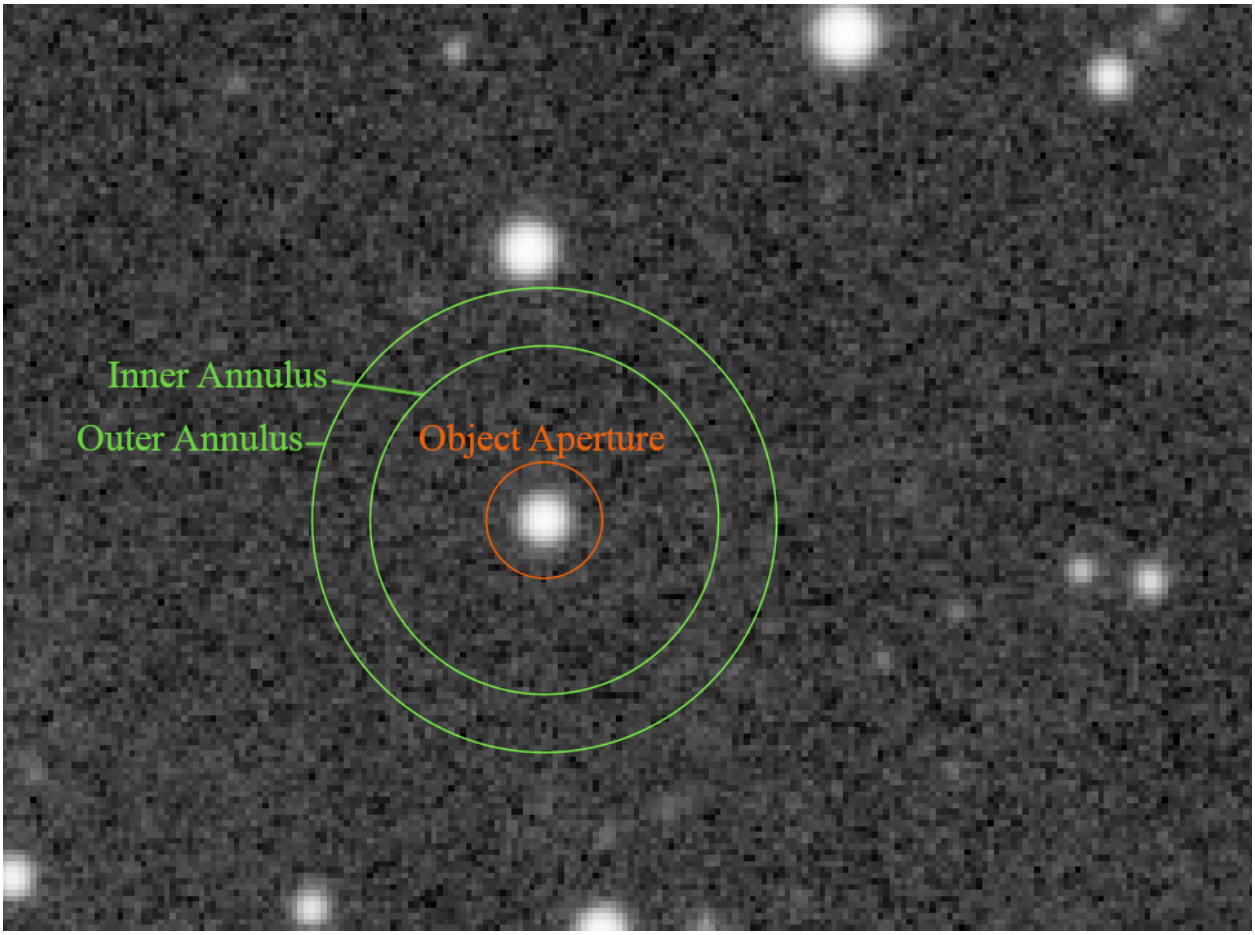


Fig. 2 – Cercles de la photométrie d'ouverture

Les valeurs de ces paramètres peuvent être modifiées dans la section *Photométrie* des préférences ou en utilisant la commande *setphot*. L'*ouverture* doit contenir tous les pixels de l'objet à mesurer, l'*annulus* ne doit par opposition contenir aucun de ses pixels. Par défaut, l'*ouverture* est ajustée pour une cible en utilisant deux fois le FWHM de la PSF, mais la taille de l'anneau est fixe. Ces valeurs doivent être ajustées pour un échantillonnage donné et vérifiées avec soin.

Note

Le texte suivant est une copie tronquée et modifiée de l'excellente documentation du logiciel MuniPack, de David Motl et publiée sous la licence de documentation libre GNU, dont les sources sont disponibles ici.

Mesurer la magnitude d'un objet

La somme *S* des pixels dans une petite zone *A* autour d'un objet est la somme de l'intensité nette de l'objet *I* et du l'intensité de l'arrière plan $B \cdot A$:

$$S = I + B \cdot A \tag{16.1}$$

La valeur de *S* et *B* est obtenu depuis l'image source, la zone *A* est déterminé comme l'aire d'un cercle de rayon *r*, avec *r* la taille de l'ouverture en pixels. Il est alors facile de calculer l'intensité net *I* d'un objet en ADU :

$$I = S - B \cdot A \tag{16.2}$$

Supposons que l'intensité net *I* est proportionnel au flux observé *F*, nous pouvons déduire la magnitude apparente *m* de l'objet en utilisant la loi de Pogson :

$$m = -2.5 \log_{10} \left(\frac{I}{I_0} \right) \tag{16.3}$$

Estimation de l'erreur de mesure

Une fois que nous avons obtenu la luminosité instrumentale brute d'un objet, nous allons essayer d'estimer son erreur standard. Tout d'abord, nous allons rappeler quelques règles générales qui s'appliquent à l'erreur standard et à sa propagation. Il s'agit d'une règle générale pour la propagation de l'erreur à travers une fonction *f* de valeur incertaine *X* :

$$\operatorname{Var}(f(X)) = \left(\frac{df}{dx} \right)^2 \operatorname{Var}(X) \tag{16.4}$$

En utilisant cette règle générale, nous déduisons deux lois de propagation des erreurs. Dans le premier cas, la valeur incertaine *X* est multipliée par une constante *a* et décalée par un décalage constant *b*. Cette loi peut également être utilisée dans le cas où il n'y a qu'une multiplication ou qu'un décalage.

$$\operatorname{Var}(aX + b) = a^2 \operatorname{Var}(X) \tag{16.5}$$

La deuxième loi définit l'erreur d'un logarithme d'une valeur incertaine *X* :

$$\operatorname{Var}(\log(\pm bX)) = \frac{\operatorname{Var}(X)}{\bar{X}^2} \tag{16.6}$$

Veuillez noter que la fonction *log* est ici le logarithme naturel, alors que la formule de Pogson (voir ci-dessus) incorpore le logarithme en base 10. L'équation suivante nous aide à gérer cette différence :

$$\log_b(x) = \frac{\log_k(x)}{\log_k(b)} \tag{16.7}$$

En mettant ces deux équations ensemble, on obtient :

$$\operatorname{Var}(\log_{10}(\pm bX)) = \frac{\operatorname{Var}(X)}{\bar{X}^2 \log(10)^2} \tag{16.8}$$

Si nous avons deux variables incertaines non corrélées *X* et *Y*, la variance de leur somme est la somme de leurs variances, cette équation est connue sous le nom de formule de Bienaymé.

$$\operatorname{Var}(X + Y) = \operatorname{Var}(X) + \operatorname{Var}(Y) \tag{16.9}$$

À partir de cette formule, nous pouvons également déduire l'erreur standard d'une moyenne d'échantillon. Si nous avons *N* observations de la variable aléatoire *X* avec une estimation basée sur l'échantillon de l'erreur standard de la population *s*, alors l'erreur standard d'une estimation de la moyenne de la population par la moyenne de l'échantillon est la suivante

$$SE_{\bar{X}} = \frac{s}{\sqrt{N}} \tag{16.10}$$

Forts de ces connaissances, nous pouvons commencer à réfléchir à l'estimation de l'erreur standard de la luminosité de l'objet. Nous considérerons les trois sources d'incertitude suivantes : (1) le bruit aléatoire à l'intérieur de l'ouverture de l'étoile qui comprend le bruit thermique du détecteur, le bruit de lecture de l'amplificateur de signal et du convertisseur analogique-numérique, (2) les statistiques de Poisson du comptage d'événements discrets (photons incidents sur un détecteur) qui se produisent pendant une période de temps fixe et (3) l'erreur d'estimation du niveau moyen du ciel.

Pour l'estimation du niveau moyen du ciel, nous avons utilisé l'algorithme de la moyenne robuste. Il permet d'estimer la variance de l'échantillon σ^2_{pxl}. Il s'agit d'une variance basée sur les pixels et comme nous avons additionné A pixels dans l'ouverture de l'étoile, la formule de Bienaymé s'applique, la somme S est une somme de A variables aléatoires non corrélées, chacune ayant une variance σ^2_{pxl}. Pour la variance de la première source d'erreur, nous obtenons :

$$\sigma_1^2 = A\,\sigma^2_{pxl} \tag{16.11}$$

où A est le nombre de pixel dans l'ouverture de l'étoile.

À partir des statistiques de Poisson, nous pouvons déduire une variance due au comptage d'événements discrets, les photons incidents sur un détecteur, qui se produisent au cours d'une période de temps fixe, l'exposition. Nous devrons à nouveau utiliser le gain p du détecteur pour convertir une intensité en ADU en un nombre de photons. Si l'intensité nette mesurée d'un objet est I, nous calculons le nombre moyen de photons λ comme suit

$$\lambda = I\,p \tag{16.12}$$

Note

La valeur du gain p du détecteur peut être modifiée dans la section *Photométrie* des préférences de Siril

Alors, la variance de l'intensité due à la statistique de Poisson est égale à sa valeur moyenne.

$$\sigma^2_{ph} = \mathrm{Var}(\mathrm{Pois}(\lambda)) = \lambda = I\,p \tag{16.13}$$

La variance est en photon, nous devons la reconvertir en ADU pour avoir la variance en ADU^2.

$$\sigma_2^2 = \frac{\sigma^2_{ph}}{p^2} = \frac{I\,p}{p^2} = \frac{I}{p} \tag{16.14}$$

Nous avons dérivé le niveau du ciel comme une moyenne d'échantillon de la population de pixels dans l'anneau du ciel. Comme chaque pixel de l'anneau a une variance σ^2_{pxl}, la variance de la moyenne de l'échantillon est de

$$s^2_{sky} = \frac{\sigma^2_{pxl}}{n_{sky}} \tag{16.15}$$

avec n_{sky} le nombre de pixel dans l'anneau de ciel.

Grâce à l'équation (16.9) nous calculons la variance de l'intensité d'un objet comme

$$\sigma^2_{ADU} = \sigma_1^2 + \sigma_2^2 + A^2\,s^2_{sky} \tag{16.16}$$

Notez que dans l'équation (16.2) le niveau du ciel est multiplié par A, nous devons donc multiplier sa variance par A^2 - voir l'équation (16.16). Maintenant, nous utilisons la loi de propagation de l'erreur pour le logarithme adopté pour correspondre à la formule de la loi de Pogson.

$$\sigma^2_{mag} = \left(\frac{-2.5}{I\,\log(10)}\right)^2 \sigma^2_{ADU} \tag{16.17}$$

En combinant les équations (16.17) et (16.16), nous pouvons dériver l'erreur standard de la luminosité de l'objet en magnitudes comme suit

$$\sigma_{mag} = \frac{1.08574}{I}\sqrt{\sigma^2_{ADU}} \tag{16.18}$$

16.2 Photométrie rapide

16.2.1 Explication des zones de photométrie

Siril n'effectuant que de la photométrie d'ouverture, il est important de comprendre et de définir les bons paramètres.

Les étoiles sont modélisées selon votre choix : Gaussienne ou Moffat comme décrit*ici* dans le chapitre*PSF dynamique*.

Ainsi, une étoile particulière sur une image particulière a sa propre FWHM.

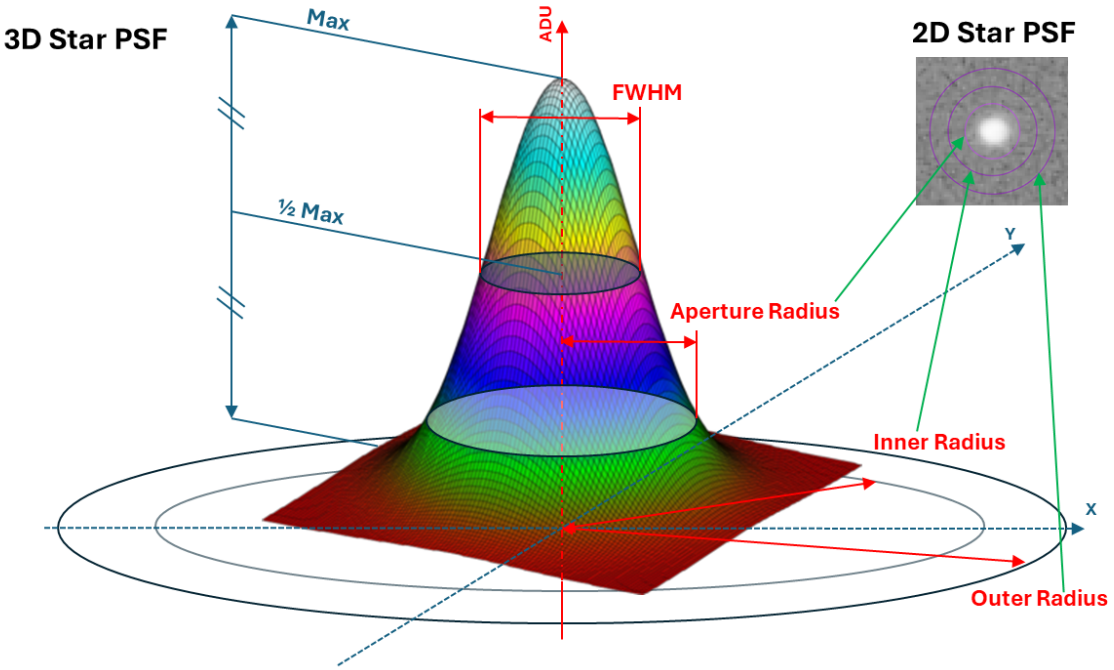


Fig. 3 – Termes relatifs à la PSF et à la photométrie.

Le processus de photométrie d'ouverture nécessite 3 rayons :

1. Le cercle extérieur, défini par le **rayon extérieur** (le plus souvent exprimé en pixels).
2. Le cercle intérieur, défini par le **rayon intérieur** (le plus souvent exprimé en pixel). Ces 2 cercles définissent l'anneau qui sert à mesurer le fond de ciel (c'est-à-dire le niveau du ciel + le bruit additionnel).
3. Le cercle d'ouverture, défini par le **rayon d'ouverture**. Il peut être exprimé en pixel ou comme un ratio de la FWHM de l'étoile (la case à cocher automatique dans la fenêtre). Cette zone est utilisée pour mesurer le signal de l'étoile ajouté au signal du fond.

Ces 3 rayons peuvent être réglés et ajustés individuellement à partir de l'interface graphique*Preferences →Photometry*.

Ou via l'interface de ligne de commande :

Ligne de commande Siril

```
setphot [-inner=20] [-outer=30] [-aperture=10] [-dyn_ratio=4.0] [-gain=2.3] [-min_
↪val=0] [-max_val=60000]
```

Obtient ou définit les paramètres de photométrie, principalement utilisés par SEQPSF. Si des arguments sont

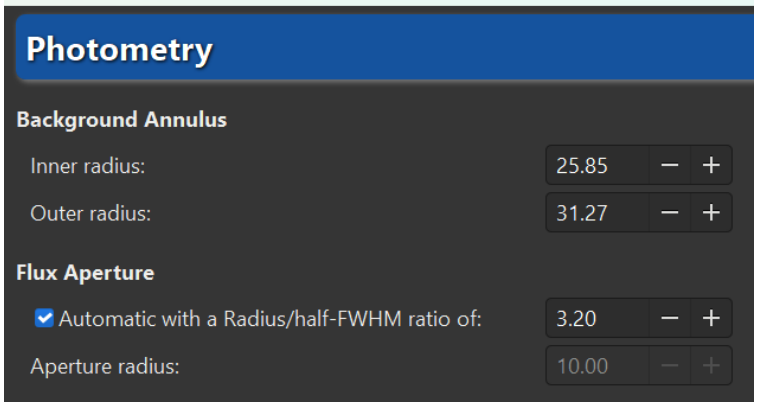


FIG. 4 – Paramètres des rayons.

fournis, ils mettront à jour les paramètres. Aucun n'est obligatoire, tous peuvent être fournis, les valeurs par défaut sont indiquées dans la syntaxe de la commande. À la fin de la commande, la configuration active sera imprimée.

La taille de l'ouverture est dynamique, sauf si elle est forcée. Dans ce cas, la valeur d'ouverture **aperture** des paramètres est utilisée. Si elle est dynamique, le rayon de l'ouverture est défini par le rapport dynamique fourni ("rayon/demi-FWHM").
Les valeurs autorisées pour l'argument **-dyn_ratio** sont comprises dans l'intervalle [1.0, 5.0]. Une valeur en dehors de cette plage fixera automatiquement l'ouverture à la valeur fixe **-aperture**.

Le gain n'est utilisé que s'il n'est pas disponible dans l'en-tête FITS

Liens : *seqpsf*

Astuce

Ces paramètres de rayon s'appliquent à tous les processus de photométrie d'ouverture : **photométrie rapide** et **courbes de lumière**. Vérifiez-les soigneusement.

16.2.2 Photométrie sur des objets sélectionnés à la main sur une seule image

Le bouton **photométrie rapide** est un bouton situé dans la barre d'outils et utilisé pour effectuer une photométrie des étoiles, c'est généralement la manière la plus simple de procéder.

Astuce

Si l'étoile se trouve au milieu de plusieurs étoiles et que l'outil ne parvient pas à pointer vers la bonne étoile, une solution alternative consiste à dessiner une sélection autour de l'étoile, puis à faire un clic droit et à cliquer sur *PSF*. Il peut aussi être intéressant de savoir que `Ctrl`-middle click (ou `Cmd`-middle click sous MacOS) dessine une sélection de taille recommandée pour la PSF/photométrie (basée sur le rayon extérieur configuré).

Astuce

Lorsque la photométrie est effectuée sur la couche RVB, les résultats sont en fait calculés sur la couche verte. Pour obtenir une photométrie sur les couches rouge ou bleue, il faut travailler sur les canaux correspondants.

Ligne de commande Siril

```
psf [channel]
```

Effectue une PSF (Fonction d'étalement du point) sur l'étoile sélectionnée et affiche les résultats. Pour les opérations en mode console, la sélection peut être donnée en pixels en utilisant BOXSELECT. S'il est fourni, l'argument **canal** sélectionne le canal de l'image sur lequel l'étoile sera analysée. Il peut être omis pour les images monochromes ou lorsqu'il est exécuté à partir de l'interface graphique avec l'un des canaux actifs dans la vue

Liens : *boxselect*

Cliquez sur ce bouton pour changer le mode de sélection de l'image, puis cliquez sur une étoile. La photométrie et la PSF (Point Spread Function) de l'étoile sont calculées, donnant de nombreux détails.

Deux modèles sont utilisés pour le calcul de la PSF, qui peuvent être sélectionnés par l'utilisateur dans la fenêtre *Dynamic-PSF* (`Ctrl` + `F6`).

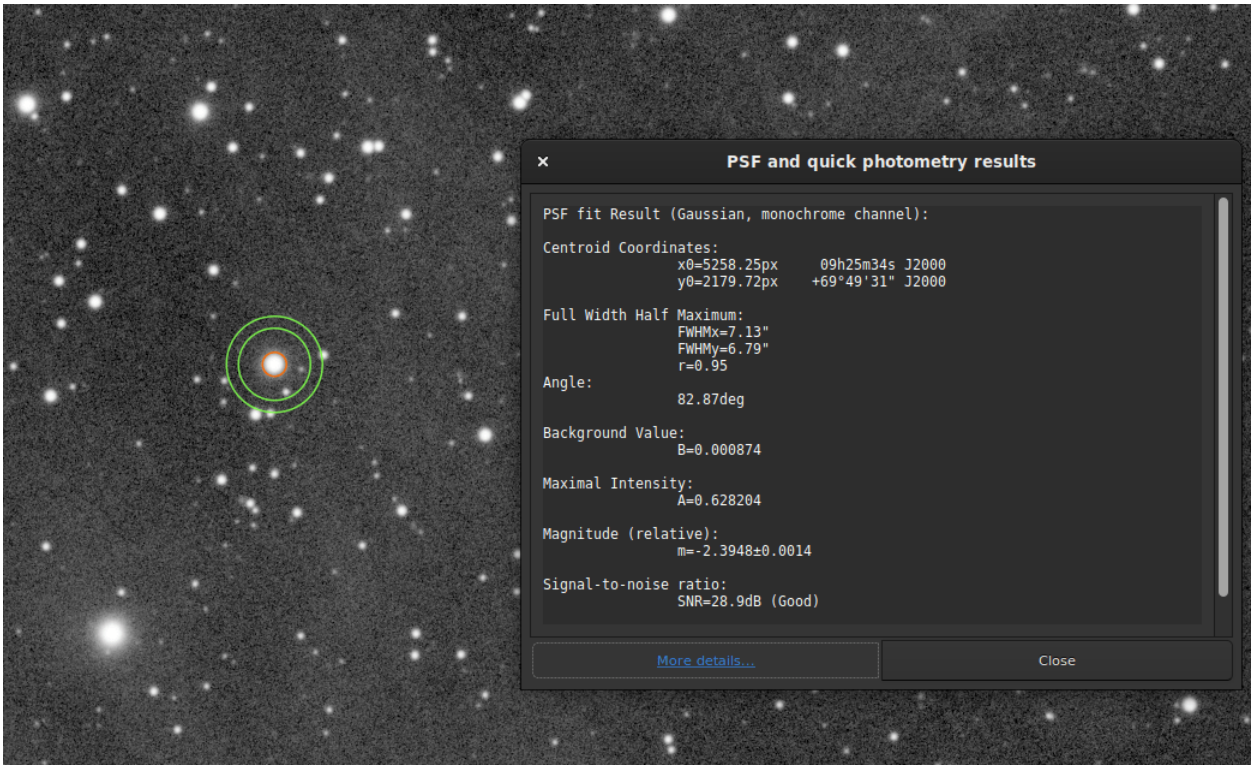


FIG. 5 – Fenêtre de résultat de la photométrie.

Le résultat de la photométrie et la PSF associée sont affichés dans le modèle :

```
PSF fit Result (Gaussian, monochrome channel):

Centroid Coordinates:
            x0=5258.25px     09h25m34s J2000
            y0=2179.72px    +69°49'31" J2000

Full Width Half Maximum:
            FWHMx=7.13"
            FWHMy=6.79"
            r=0.95
Angle:
            82.87deg

Background Value:
            B=0.000874

Maximal Intensity:
            A=0.628204

Magnitude (relative):
            m=-2.3948±0.0014

Signal-to-noise ratio:
            SNR=28.9dB (Good)

RMSE:
            RMSE=1.890e-03
```

1. L'ajustement a été effectué avec la fonction d'ajustement **Gaussien**, aucun paramètre supplémentaire n'est donc nécessaire. Cependant, si Moffat a été utilisé, la sortie suivante sera affichée :

   ```
   PSF fit Result (Moffat, beta=2.9, monochrome channel):
   ```

2. **Les coordonnées du centroïde** donnent les coordonnées du centroïde en pixels. Cependant, comme dans l'exemple ci-dessus, si l'astrométrie a été définie sur l'image, Siril donne les coordonnées dans les Systèmes de coordonnées mondiales (RA et Dec).
3. La **largeur à mi-hauteur** (FWHM) est retournée en arcsec si l'échelle de l'image est connue (obtenue à partir de son en-tête ou de l'interface graphique :menuselection : *Information de l'image* --> *Information*) et en pixels si ce n'est pas le cas. L'arrondi r est également calculé comme le ratio de $\frac{\text{FWHMy}}{\text{FWHMx}}$.
4. **L'angle** est l'angle de rotation de l'axe X par rapport aux coordonnées du centroïde. Il varie dans l'intervalle $[-90, +90]$.
5. **La valeur de l'arrière-plan** est l'arrière-plan local dans la plage $[0, 1]$ pour les images à 32 bits et $[0, 65535]$ pour les images à 16 bits. Il s'agit d'une valeur ajustée, et non de l'arrière-plan calculé dans l'anneau de photométrie d'ouverture.
6. La valeur **intensité maximale** est également une valeur ajustée et représente l'amplitude. Il s'agit de la valeur maximale de la fonction ajustée, située aux coordonnées du centroïde.
7. La **magnitude**, donnée avec son incertitude, est le résultat de la photométrie. Toutefois, si pour une raison quelconque le calcul ne peut être effectué (pixels saturés ou noirs), une incertitude de **9,999** est donnée. Dans ce cas, la photométrie est signalée comme invalide mais une valeur de magnitude est tout de même donnée, bien qu'elle doive être utilisée avec précaution.
8. Un estimateur du **rapport signal/bruit** est indiqué dans les résultats. Sa valeur est calculée à partir de la formule

suivante et indiquée en dB :

$$\mathrm{SNR} = 10 \log_{10} \left(\frac{I}{N} \right) \tag{16.19}$$

où I est l'intensité nette, proportionnelle au flux observé F et N le total des incertitudes comme exprimé dans (16.18).

Pour une meilleure compréhension, il est associé à 6 niveaux de qualité :

1. Excellent (SNR > 40dB)
2. Bon (SNR > 25dB)
3. Satisfaisant (SNR > 15dB)
4. Faible (SNR > 10dB)
5. Mauvais (SNR > 0dB)
6. N/A

   Cette dernière notation n'est affichée que si le calcul a échoué, pour une raison ou une autre.

9. Enfin, **RMSE** donne un estimateur de la qualité de l'ajustement. Plus la valeur est faible, meilleur est le résultat.

Lorsque l'image est résolue, le bouton *Plus de détails* en bas de la fenêtre renvoie à une page du site SIMBAD contenant des informations sur l'étoile sélectionnée. Cependant, il est possible que la page ne donne aucune information supplémentaire si l'étoile n'est pas dans la base de données SIMBAD.

FIG. 6 – Plus de détails à propos de l'étoile avalisée. Cliquez sur limage pour l'agrandir.

16.2.3 Photométrie rapide sur les séquences

Une photométrie rapide peut également être effectuée sur une séquence. Ceci est généralement destiné à obtenir une courbe de lumière comme expliqué *ici*. Pour cela, il faut **charger une séquence**, faire une sélection autour d'une étoile, puis **clic droit** sur l'image.

> **Astuce**
>
> Idéalement, la séquence doit être alignée sans interpolation afin de ne pas altérer les données brutes. Par exemple, utilisez **l'alignement global** avec l'option **Enregistrer la transformation dans le fichier seq uniquement**.

> **Note**
>
> Assurez-vous que les rayons intérieurs et extérieurs de l'anneau d'arrière-plan sont adaptés à l'étoile et à la séquence analysées. Certaines images peuvent avoir des FWHM beaucoup plus grandes que l'image de référence, à cause des conditions du ciel ou d'un mauvais suivi. Elles peuvent être modifiées dans les *préférences* ou avec la commande `setphot`.

A la fin du processus, Siril ouvre automatiquement l'onglet de tracé montrant les courbes calculées. Il est possible de cliquer sur plusieurs étoiles pour reproduire le calcul, cependant la première étoile garde le statut particulier de *variable*, et les autres servent de *références*. Ceci est important pour le calcul de la courbe de lumière.

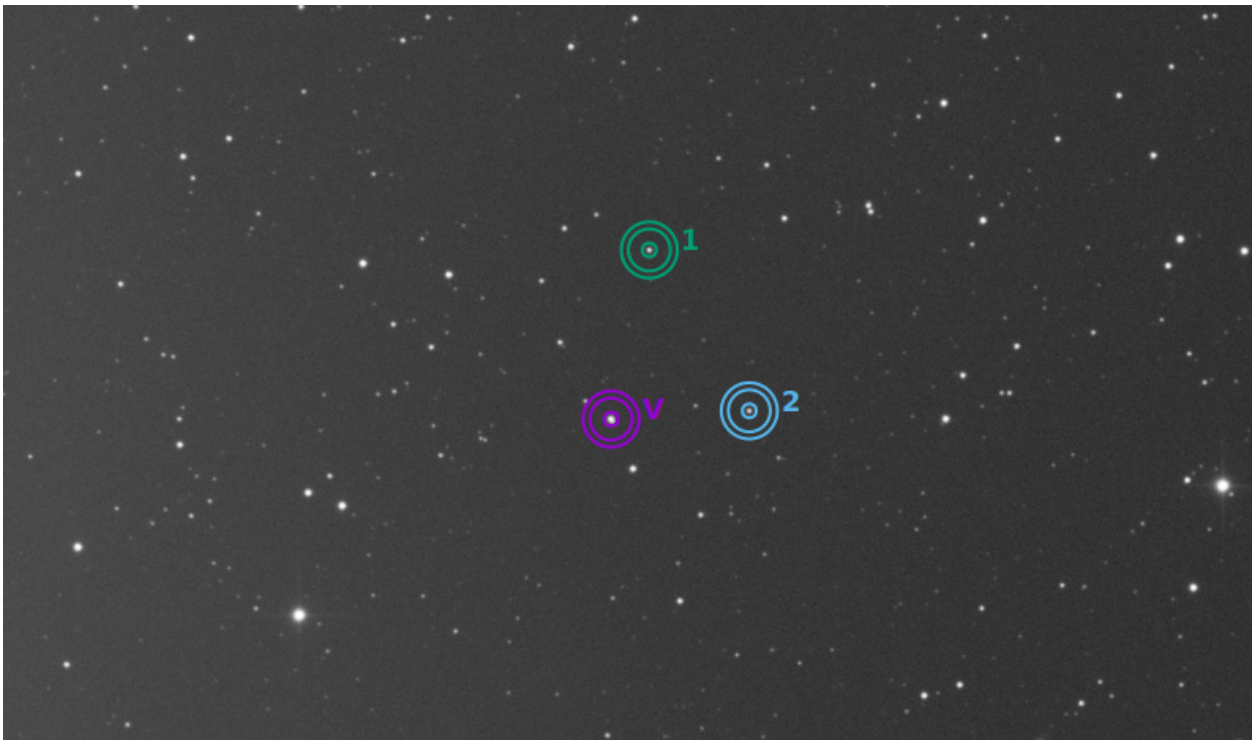


Fig. 7 – Dans cet exemple, 3 étoiles ont été analysées La première est utilisée comme variable. Les autres sont les références.

16.2.4 Calcul des magnitudes réelles

La magnitude calculée n'a de sens que si elle est comparée aux autres dans l'image linéaire. En effet, la valeur donnée ne correspond pas du tout à la véritable magnitude visible de l'étoile, elle est non calibrée, on parle aussi de magnitude relative.

Siril propose des outils permettant de calculer une magnitude apparente approximative. Pour cela, il faut connaître la magnitude d'une autre étoile visible sur l'image qui servira de référence. Il n'est actuellement possible d'utiliser qu'une seule étoile comme référence, d'où le terme *approximative*. Pour une plus grande précision, il faut choisir une étoile de couleur et de magnitude similaires à celles de l'étoile ou des étoiles que l'on veut mesurer, et sa magnitude fournie doit être en adéquation avec le filtre utilisé pour capturer l'image. Les catalogues contiennent des magnitudes calculées à l'aide d'un filtre photométrique, qui n'est généralement pas ce que les amateurs utilisent pour faire de belles images, ce qui ajoute une autre approximation.

— Faites une photométrie rapide sur une étoile connue, la magnitude relative donnée est `-2.428`. Il est possible de trouver la magnitude visible réelle en cliquant sur le bouton *Plus de détails* comme expliqué ci-dessus.

Disons que la valeur trouvée est `11.68` (assurez-vous d'utiliser une valeur correspondant à la bande spectrale de l'image).

— Une fois fait, conservez l'étoile sélectionnée, puis entrez la commande suivante dans Siril

```
setmag 11.68
```

Cela produira quelque chose comme

```
10:50:49: Relative magnitude: -2.428, True reduced magnitude: 11.680, Offset:␣
↪14.108
```

Ligne de commande Siril

```
setmag magnitude
```

Étalonne les magnitudes en sélectionnant une étoile et en donnant la magnitude apparente connue.

Tous les calculs de PSF renverront ensuite la magnitude apparente étalonnée, au lieu d'une magnitude apparente relative aux valeurs ADU. Notez que la valeur fournie doit correspondre à la magnitude du filtre d'observation pour être significative.
Pour réinitialiser la constante de magnitude voir UNSETMAG

Liens : *psf*, *unsetmag*

— Maintenant, toutes les magnitudes calculées doivent avoir des valeurs proches de leur magnitude visuelle réelle. Cependant, ceci est particulièrement vrai pour les étoiles dont la magnitude est du même ordre de grandeur que l'étoile prise comme référence.

— Pour annuler le décalage calculé, il suffit de taper

```
unsetmag
```

Ligne de commande Siril

```
unsetmag
```

Remet l'étalonnage de la magnitude à 0. Voir SETMAG

Liens : *setmag*

Astuce

Les mêmes commandes existent pour les séquences. Ce sont `seqsetmag` et `sequnsetmag`. Elles sont utilisées de la même manière lorsqu'une séquence est chargée.

Ligne de commande Siril

```
seqsetmag magnitude
```

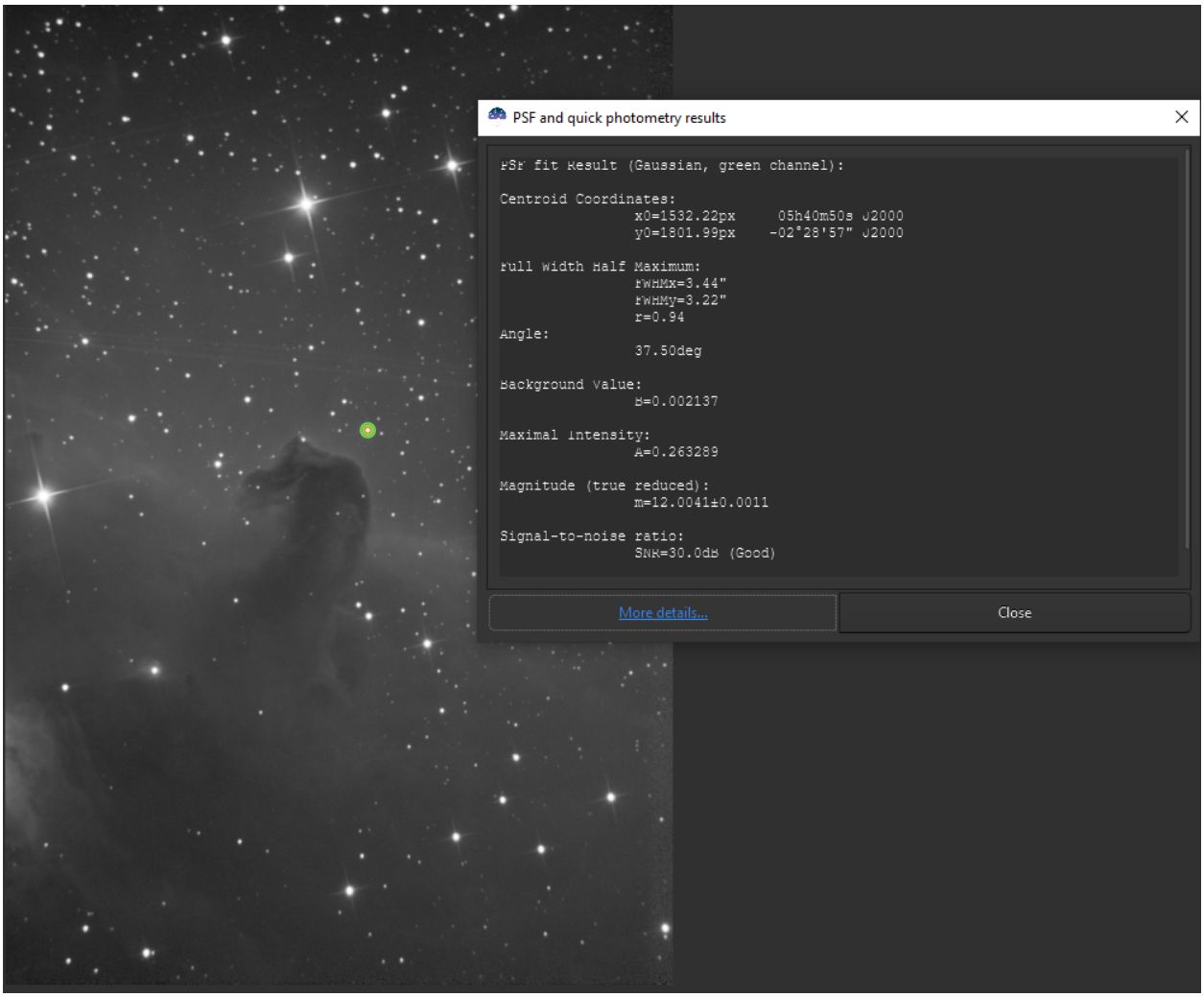


FIG. 8 – Fenêtre de résultat de la photométrie avec la magnitude réelle définie.

Identique à la commande SETMAG mais pour la séquence chargée.

Cette commande n'est valable qu'après avoir exécuté SEQPSF ou sa contrepartie graphique (sélectionner la zone autour d'une étoile et lancer l'analyse PSF de la séquence, elle apparaîtra dans les graphiques).
Cette commande a le même but que SETMAG mais recalcule la magnitude de référence pour chaque image de la séquence où l'étoile de référence a été trouvée.
Lors de l'exécution de la commande, la dernière étoile qui a été analysée sera considérée comme l'étoile de référence. L'affichage du graphe de magnitude avant de taper la commande facilite sa compréhension.
Pour réinitialiser l'étoile de référence et le décalage de magnitude, voir SEQUNSETMAG

Liens : *setmag*, *seqpsf*, *psf*, *sequnsetmag*

Ligne de commande Siril

```
sequnsetmag
```

Réinitialise la calibration de la magnitude et l'étoile de référence pour la séquence. Voir SEQSETMAG

Liens : *seqsetmag*

16.3 Courbe de lumière

En astronomie, une courbe de lumière est un graphique de l'intensité lumineuse d'un objet céleste en fonction du temps, généralement avec la magnitude de la lumière reçue sur l'axe y et le temps sur l'axe x. Siril est capable de générer de telles courbes lors de l'analyse des étoiles.

Il y a maintenant deux façons de sélectionner les étoiles variables et les étoiles de référence (également appelées comparaisons) : manuellement, ou en utilisant une liste d'étoiles obtenue par le plugin exoplanète du N.I.N.A..

16.3.1 Détection d'étoile manuelle

Commencez par passer en **mode photométrie** (cliquez sur le bouton).

Sélectionnez (clic droit) l'étoile variable suivie des étoiles de comparaison que vous souhaitez, comme expliqué*ici*.

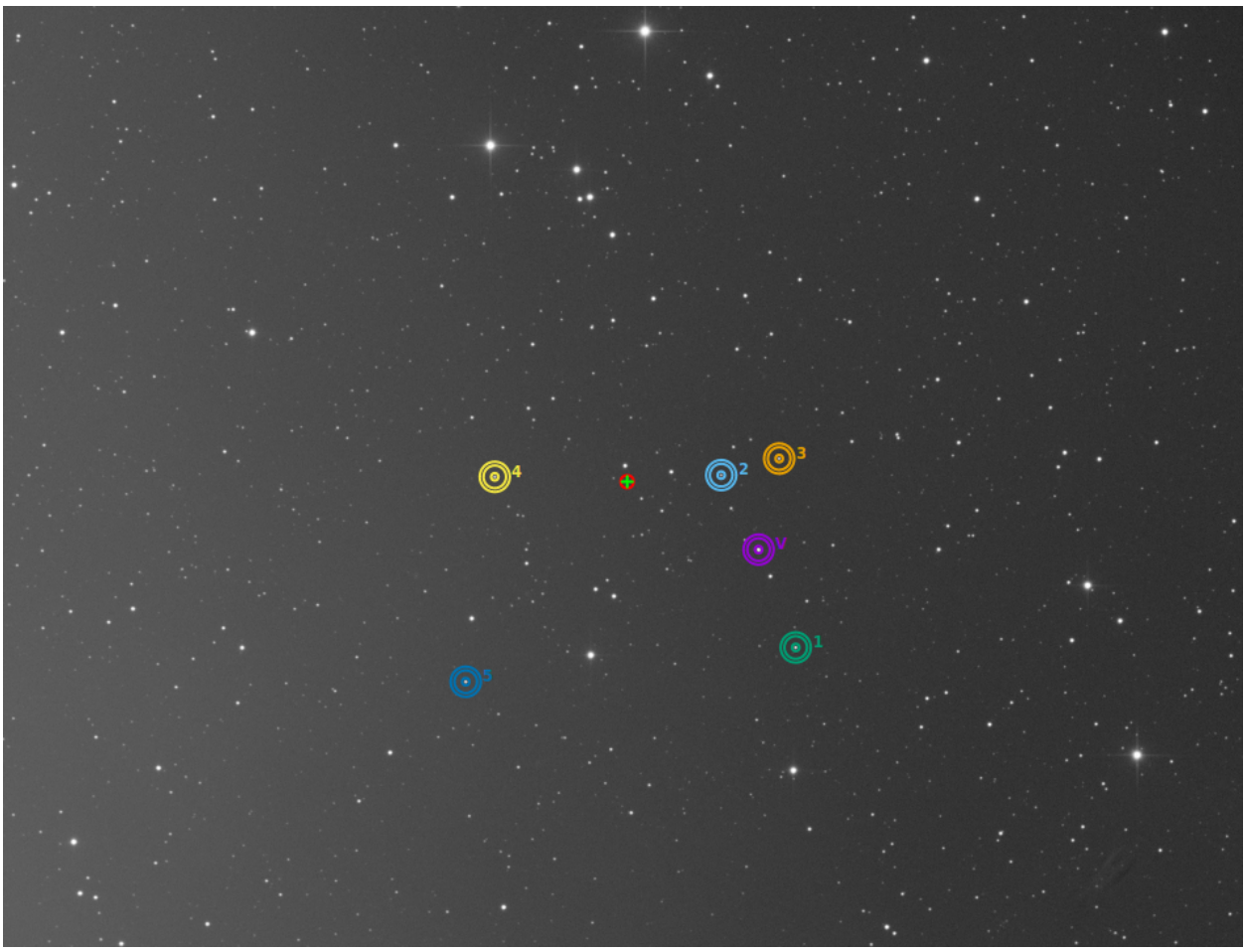


FIG. 9 – Une étoile est variable (violette avec un V) et les 5 autres sont utilisés comme références.

Avertissement

Veillez à ne pas sélectionner d'étoiles variables comme références. Si l'astrométrie est faite sur votre image, n'hésitez pas à utiliser la *requête SIMBAD* pour en savoir plus sur les étoiles.

Un moyen pratique de vérifier votre étoile de comparaison est d'afficher les 3 catalogues d'étoiles variables : GCVS, AAVSO-VSX et GAIA-varisum.

Vous devez définir votre propre paramètre (magnitude limite), mais à titre d'exemple, voici les commandes :

```
conesearch 18.0 -cat=gcvs
conesearch 18.0 -cat=aavso_chart
conesearch 18.0 -cat=varisum
```

Astuce

Il est préférable de choisir des références dont la magnitude est proche de celle de la variable comme expliqué plus loin*ici*.

Une fois réalisé, Siril lance automatiquement l'onglet Graphique tel que montré dans la figure ci dessous. Ceci montre la courbe de la FWHM en fonction du numéro de l'image.

Ce qui nous intéresse dans cette partie est d'afficher les courbes de magnitude. Il suffit d'aller dans le menu déroulant et de changer **FHWM** en **Magnitude**. Les courbes de magnitude de chaque étoile analysée sont alors affichées. Les sous-menus situés dans *Photometrie* du menu *Outils* sont également activés.

Sauvegarde des étoiles de comparaison sélectionnées

Lorsque l'on est satisfait du jeu d'étoiles de comparaison, on peut l'enregistrer pour une utilisation ultérieure dans un processus*Courbe de lumière automatisée* ou simplement garder une trace pour une documentation ultérieure.

Dans le menu*Outils*, sélectionnez*Photométrie → Créer un fichier d'étoiles de comparaison...*.

L'option par défaut*Utiliser les étoiles de l'image actuellement chargée* est sélectionnée.

Le nom du fichier de sortie par défaut sera `V_SirilstarList_user.csv` mais vous pouvez le modifier à votre convenance.

Après avoir cliqué sur*OK*, le fichier sera créé et sauvegardé dans votre répertoire de travail.

Courbe de lumière (format ETD)

Le bouton *Courbe de lumière (format ETD)*, situé dans le menu *Outils → Photométrie → Créer une sortie* est dédié aux données de transit d'exoplanètes, produisant un fichier dans le format géré par le site web de l'ETD.

Une fois l'analyse terminée avec un nombre d'étoiles de référence d'au moins 4 ou 5 (plus le nombre est élevé, plus le résultat est précis. Cependant, un nombre maximum de 19 étoiles est autorisé). Siril demande un nom de fichier pour enregistrer les données au format `csv`, puis la courbe de lumière est affichée dans une nouvelle fenêtre.

```
#JD_UT (+ 2457403)
# JD_UT V-C err
2457403.260428 -1.01054 0.00468598
2457403.261956 -1.00321 0.0046727
2457403.263461 -1.00132 0.00540513
2457403.264977 -0.994067 0.00505492
2457403.266481 -1.00397 0.00465623
```

(suite sur la page suivante)

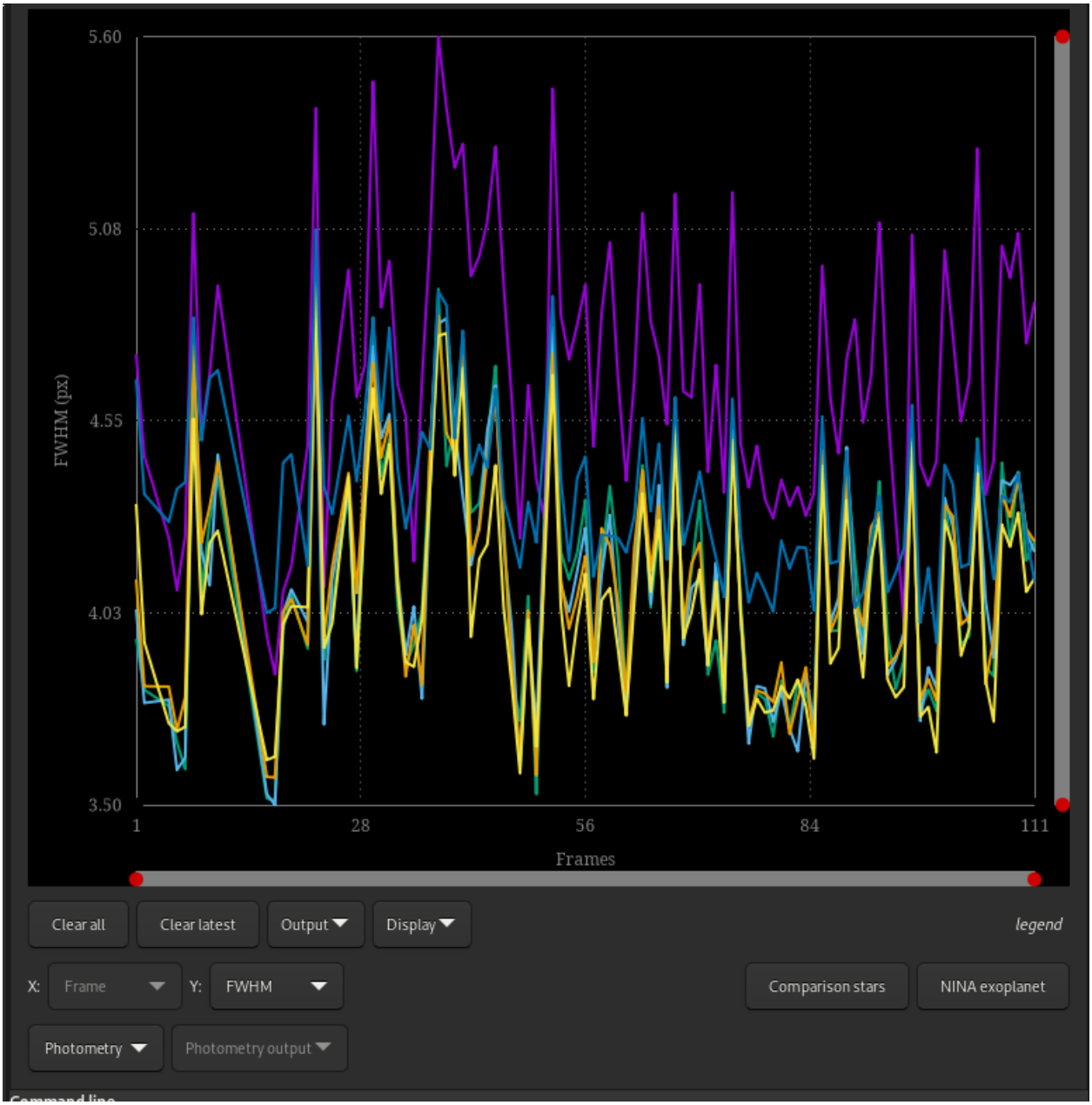


FIG. 10 – L'onglet graphique tel que montré après une photométrie rapide sur la séquence.

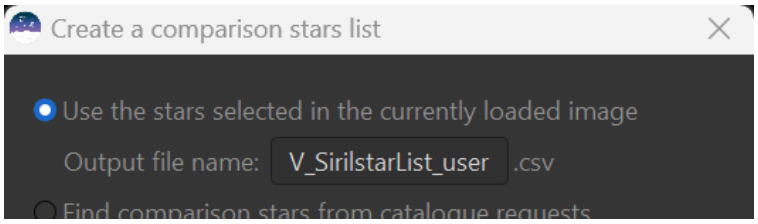


FIG. 11 – Enregistrement des étoiles sélectionnées à la main dans un fichier .csv.

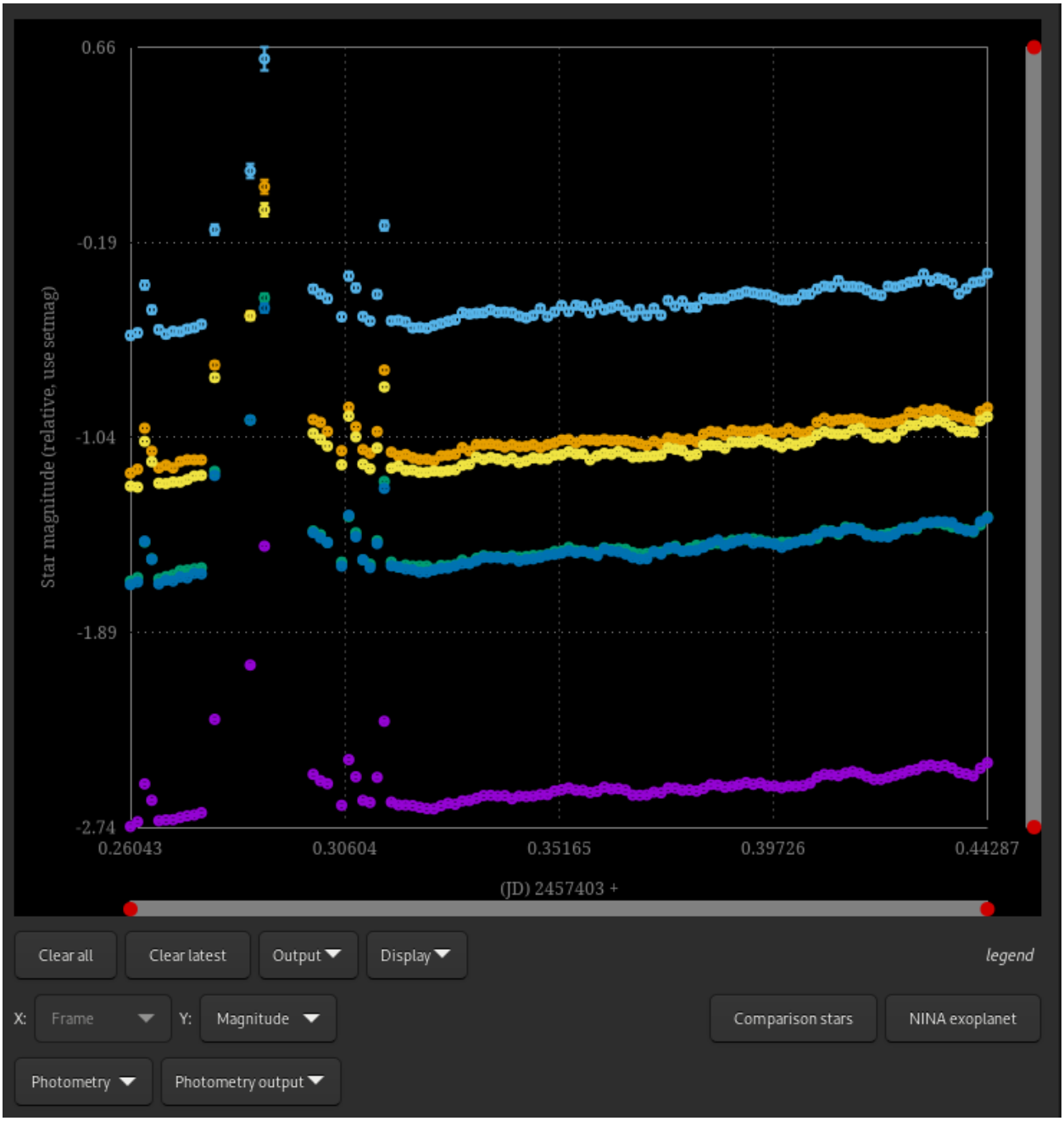


FIG. 12 – Le passage en mode magnitude rend le bouton *Résultats photométriques* sensible.

(suite de la page précédente)

```
2457403.267998 -1.00532 0.00467666
...
```

Il s'agit d'un fichier à trois colonnes. Les 2 premières lignes, commençant par un "#", sont des commentaires. Ensuite, la première colonne est l'heure précise de l'observation au format de la date julienne. La deuxième colonne est la magnitude de la variable (V), à laquelle a été soustraite la somme des magnitudes instrumentales des étoiles de comparaison. La troisième colonne représente l'incertitude sur la magnitude.

Théorie

V est donné par la formule suivante :

$$V = (V_{\text{ins}} - C_{\text{ins}}), \tag{16.20}$$

avec

$$C_{\text{ins}} = -2.5 \log_{10} \left(\frac{1}{N} \sum 10^{-0.4 C_{\text{ins},i}} \right). \tag{16.21}$$

N est le nombre total d'étoiles de comparaison et C_{ins}, i est la magnitude instrumentale de la i-ième étoile de l'ensemble.

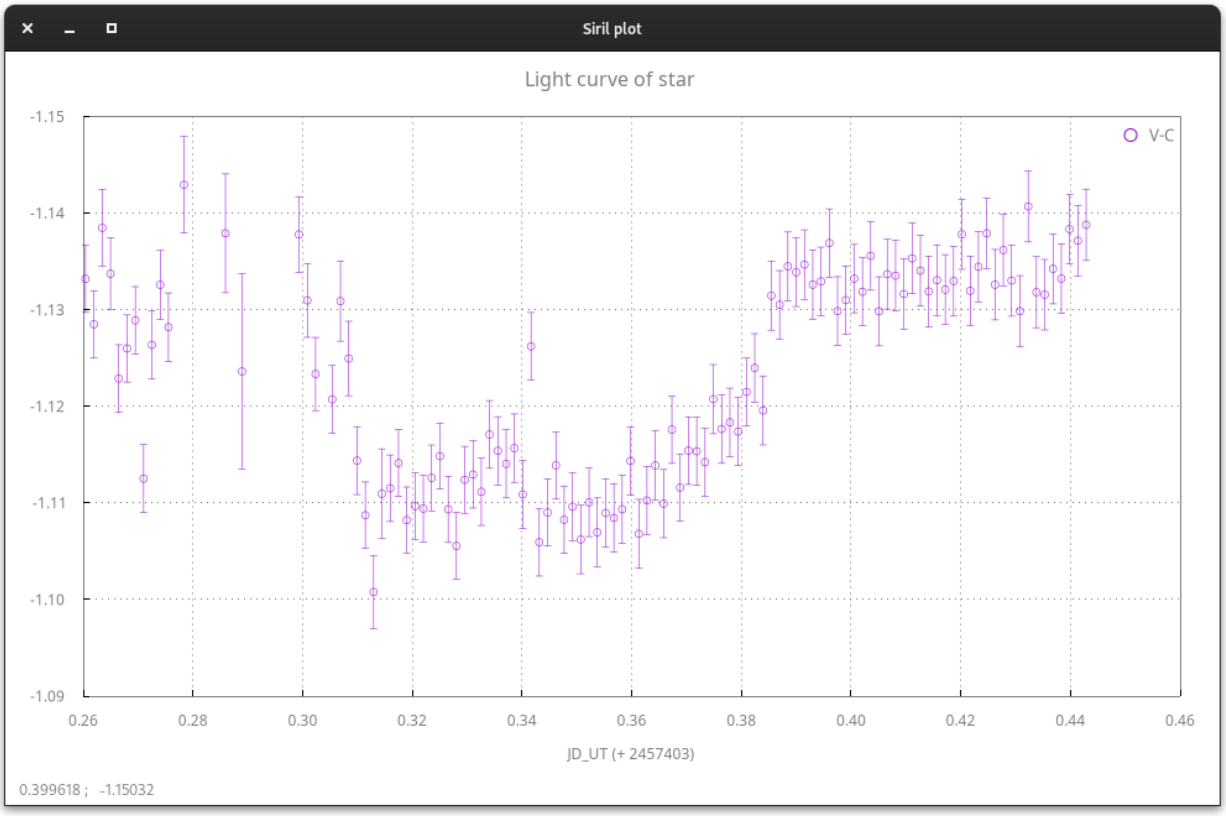


FIG. 13 – Courbe de lumière d'un transit d'exoplanète.

Astuce

La courbe de lumière est tracée à l'aide de l'outil de graphique interne de Siril. Reportez-vous à sa *documentation* pour obtenir des instructions plus détaillées sur son utilisation.

Format de fichier étendu de l'AAVSO

Situé dans le menu *Outils* → *Photométrie* → *Créer une sortie*, le deuxième bouton, *Format de fichier AAVSO*, est dédié au format de fichier étendu de l'AAVSO pour les étoiles variables. Il ouvre une nouvelle boîte de dialogue qui, une fois complétée, exporte un fichier csv qui peut être téléchargé sur le site web dédié.

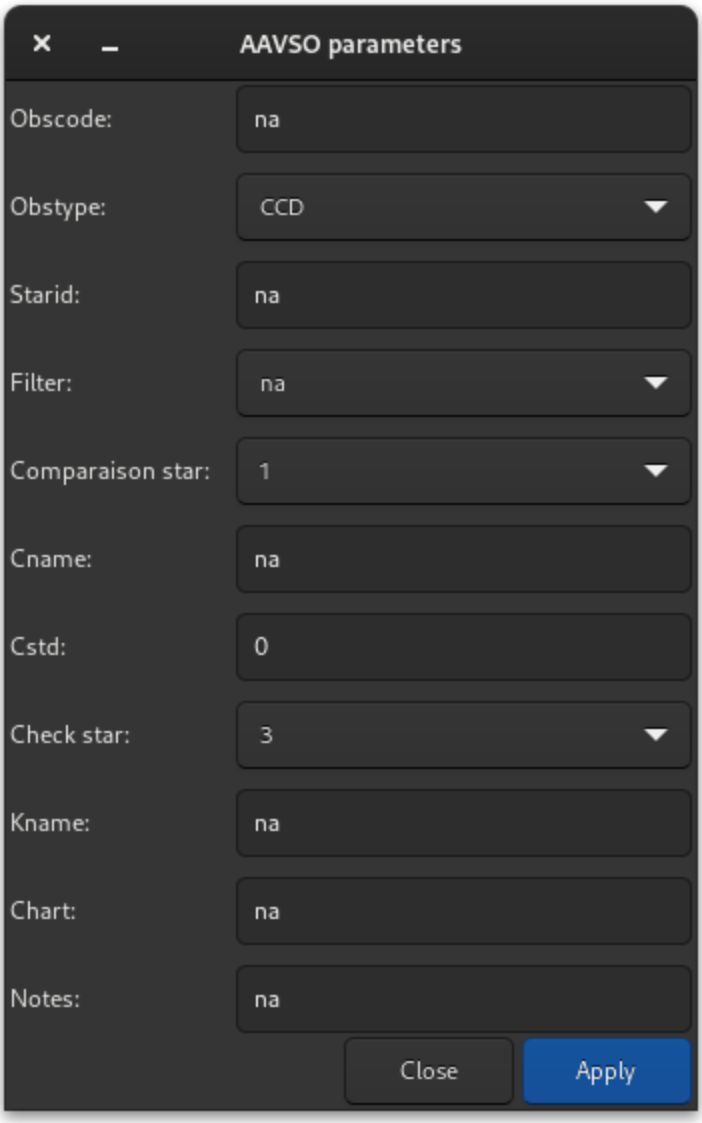


Fig. 14 – Dialogue des paramètres AAVSO. Ce dialogue est utilisé pour remplir le fichier étendu de l'AAVSO. Plus d'informations sont disponibles sur le site web de l'AAVSO

Théorie

Dans Siril, le format de fichier AAVSO exporté utilise la magnitude standardisée définie comme suit :

$$V_{\text{std}} = (V_{\text{ins}} - C_{\text{ins}}) + C_{\text{std}} \tag{16.22}$$

où V_{ins} et C_{ins} sont la magnitude instrumentale de la variable et de la comparaison, respectivement, et C_{std} la magnitude réelle pour la comparaison. Vous pouvez spécifier la valeur de C_{std} dans la boîte de dialogue des paramètres de l'AAVSO, avant l'exportation.

Un exemple de photométrie est donné ci-dessous avec l'étoile Rx And. Cette étoile présente une grande variabilité : sa magnitude varie de 10,2 à 15,1 avec une périodicité d'environ 13 jours. Après traitement dans Siril, en prenant une étoile de comparaison et une étoile de contrôle, nous obtenons la courbe AAVSO suivante après téléchargement des données. Les nouvelles données sont à l'extrême droite (prises les 24 et 25 novembre 2023) et sont représentées en vert.

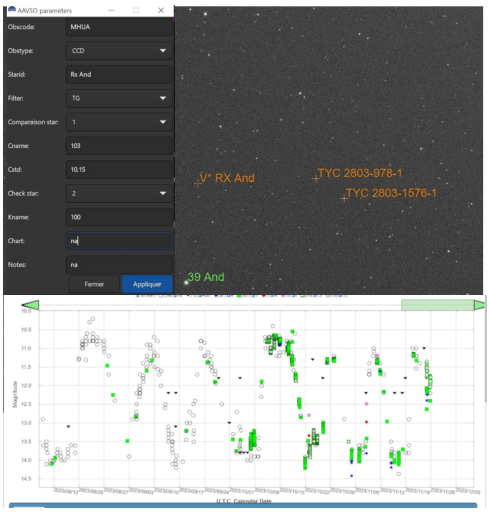


FIG. 15 – Un exemple concret de photométrie d'étoile variable (avec l'aimable autorisation de H. Meunier).

16.3.2 Sélection automatique des étoiles

Courbe de lumière automatisée

Afin d'automatiser le processus d'analyse des transits d'exoplanètes, des listes d'étoiles de référence, également appelées étoiles de comparaison, peuvent être obtenues à partir de catalogues d'étoiles, avec les critères appropriés : magnitude similaire, couleur similaire (pour ne pas modifier leur magnitude relative avec l'extinction atmosphérique à différentes altitudes), proximité.

Le logiciel de capture N.I.N.A possède un plugin exoplanète qui affichera ces étoiles et permettra d'enregistrer la liste dans un fichier CSV, comme par exemple `csv file` :

```
Type,Name,HFR,xPos,yPos,AvgBright,MaxBright,Background,Ra,Dec
Target,HD 189733 b,2.6035068712769851,1992,1446,1640.3703703703704,39440,1917.
↪0601851851852,300.18333333333328,22.709722222222222
Var,SW Vul,2.8626145609282911,2972,276,26.14,2012,1905.445,300.02171,22.93517
Var,DQ Vul,2.372369130017419,3006,1040,28.180555555555557,2048,1906.9027777777778,300.
↪01254,22.78103
Var,HQ Vul,3.8351043206620834,157,1690,49.393939393939391,2104,1905.7454545454545,300.
```

(suite sur la page suivante)

(suite de la page précédente)

```
↪55808,22.64067
...
Comp1,ATO J300.3222+22.7056,2.4268101078425852,1367,1465,352,4496,1913.9504132231405,300.
↪32229415181337,22.705681453738887
Comp1,HD 189657,2.5343988482845927,2527,2808,23.814814814814813,2012,1906.5061728395062,
↪300.08714683055996,22.4400393728
...
Comp2,000-BJP-946,2.2738807043120195,1832,750,29.962962962962962,2024,1910.0648148148148,
↪300.23741666666666,22.846999999999998
Comp2,000-BJP-942,2.0977710589704297,2760,1572,31.083333333333332,2096,1908.
↪6527777777778,300.025875,22.704777777777778
...
```

Dans le menu *Outils*, Siril peut charger ce fichier en utilisant le bouton *Courbe de lumière automatisée...*. Pour l'utiliser, quelques conditions préalables doivent être remplies :

- la séquence d'images calibrées doit être déjà chargée
- l'image de référence de la séquence doit avoir été résolue astrométriquement, pour s'assurer de l'identification correcte des étoiles dans les coordonnées équatoriales J2000

À partir de là, tout est automatique et la courbe de lumière des étoiles de comparaison sélectionnées est affichée à la fin du processus.

Dans cette fenêtre, les*zones d'ouverture et d'arrière-plan* utilisées sont également rappelées.

La vidéo suivante montre un traitement automatisé de la courbe de lumière avec une liste d'étoiles de comparaison provenant de NINA :

Courbe de lumière par commande

Il est également possible d'automatiser ou de créer la courbe de lumière à distance en utilisant la commande `light_curve`. Comme l'opération en aveugle nécessite autant d'automatisation que possible, la configuration des rayons de l'anneau d'arrière-plan peut être automatisée avec l'argument `-autoring` : cela lance une détection d'étoiles dans l'image de référence et multiplie la FWHM moyenne avec un facteur configurable pour obtenir les rayons intérieurs et extérieurs qui devraient fonctionner avec la séquence.

Ligne de commande Siril

```
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 plusieurs étoiles avec photométrie d'ouverture dans une séquence d'images et produit une courbe de lumière pour l'une d'entre elles, calibrée par les autres. Les premières coordonnées, en pixels si **-at=** est utilisé ou en degrés si **-wcs=** est utilisé, sont pour l'étoile dont la lumière sera tracée, les autres pour les étoiles de comparaison.

Alternativement, une liste d'étoiles cibles et de référence peut être transmise dans le format de la liste d'étoiles du plugin exoplanète de NINA, avec l'option **-ninastars=**. Siril vérifie que toutes les étoiles de référence peuvent être utilisées avant de les utiliser. Un fichier de données est créé dans le répertoire courant nommé light_curve.dat, puis Siril trace le résultat dans une image PNG

Les rayons de l'anneau peuvent être configurés dans les paramètres ou fixés à un facteur de la FWHM de l'image de référence si l'option **-autoring** est activée. Ces tailles de rayons sont 4,2 fois et 6,3 fois le FWHM pour les rayons intérieurs et extérieurs, respectivement.

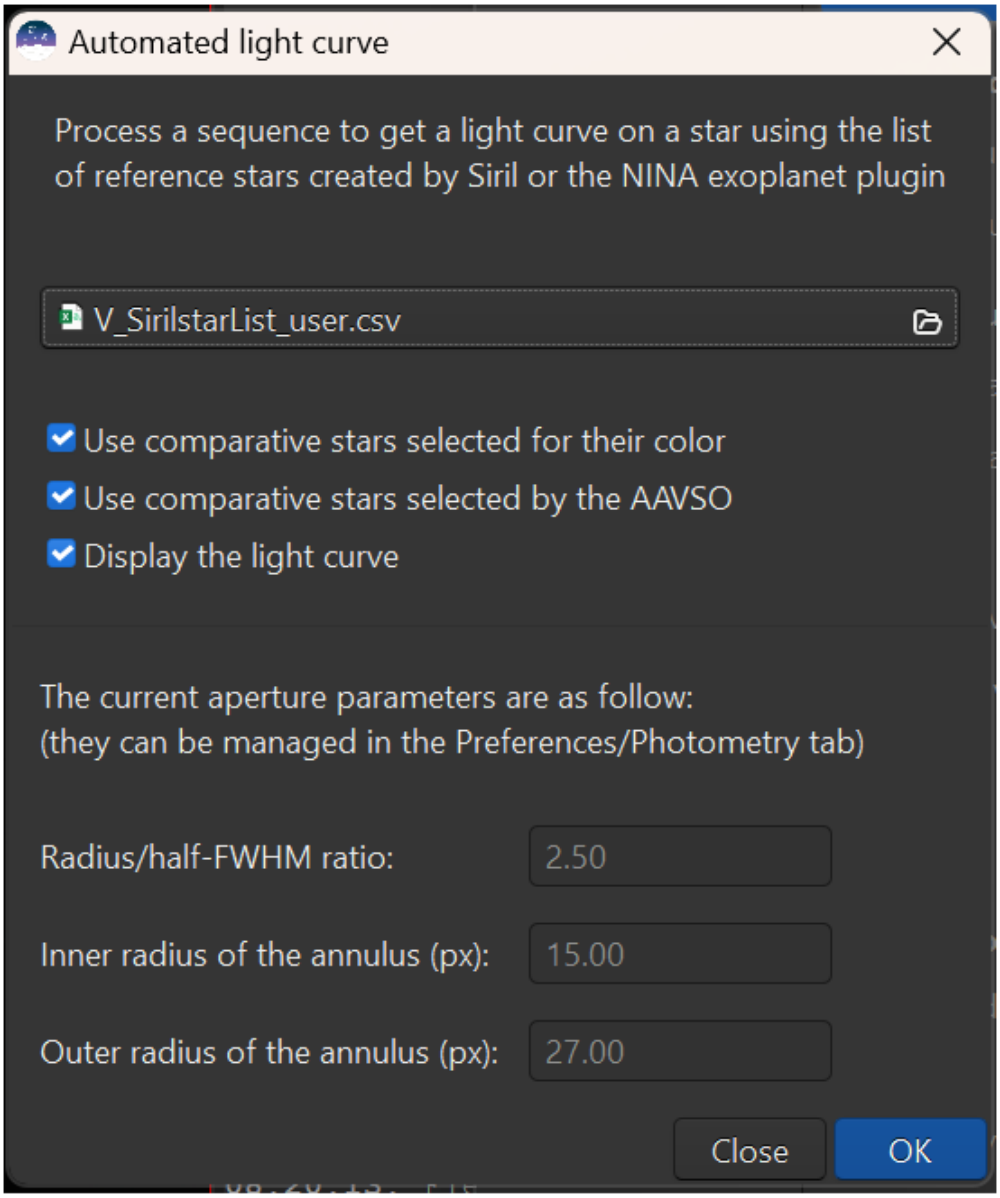


FIG. 16 – Fenêtre de la photométrie automatique.

Voir également la commande **setphot** pour définir de la même manière la taille du rayon d'ouverture.

Voir aussi SEQPSF pour les opérations sur une seule étoile

Liens : *seqpsf*

16.3.3 Générer une liste d'étoiles de comparaison

Si vous n'êtes pas un utilisateur de NINA, ou si vous souhaitez créer votre propre fichier et utiliser ensuite la fonction*Courbe de lumière automatisée*, vous pouvez, depuis Siril-1.3, générer une liste d'étoiles de comparaison à l'aide du nouvel outil Siril.

Liste via l'interface graphique

Cette fonction est accessible via le menu*Outils*,*Photométrie* → *Créer un fichier d'étoiles de comparaison...*, une fois que l'image affichée de la séquence chargée a été résolue par astrométrie.

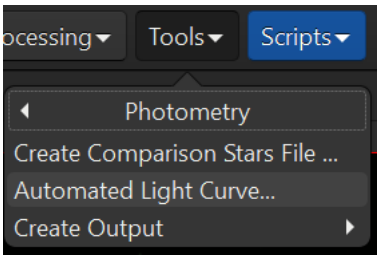


FIG. 17 – La fonction*Créer un fichier d'étoiles de comparaison...* dans le menu*Outils*.

La condition préalable est de charger une image représentative de la séquence, résolue par astrométrie.

- L'option **Trouver des étoiles de comparaison dans le catalogue** est obligatoire dans le cas d'utilisation.
- Le premier champ doit être le **nom de l'étoile variable** dont vous souhaitez obtenir la courbe de lumière.
- Le bouton à cocher **Champ de vision étroit** permet de sélectionner un cercle centré sur l'image.
- Le deuxième champ est la valeur de **delta_Vmag**. Il s'agit de l'écart maximal par rapport à la cible **magnitude V** (0,0 < delta_Vmag < 6,0). La valeur par défaut est fixée à 3,0.
- Le troisième champ est la valeur **delta_BV**. Il s'agit de l'écart maximal par rapport à l'indice **BV** cible (0,0 < delta_BV < 0,7). La valeur par défaut est fixée à 0,5.
- Le quatrième champ est la valeur **max_emag**. Il s'agit de l'erreur statistique maximale autorisée sur Vamg (0,0 < max_emag < 0,1). La valeur par défaut est de 0,03.
- Ensuite, vous pouvez choisir la source des étoiles de comparaison : du catalogue APASS ou du catalogue NOMAD.

Ces deux critères vous permettent d'obtenir des étoiles de comparaison correspondant à la magnitude et à l'indice de couleur de la cible.

Le nombre maximum d'étoiles de comparaison est fixé en interne à 20.

Après avoir cliqué sur*OK*, une requête est envoyée au serveur adéquat.

La réponse est une liste d'étoiles, provenant du catalogue choisi, correspondant à vos critères, affichée en rouge dans votre image et sauvegardée sous forme de fichier CSV compatible avec le processus Exoplanète de NINA.

Note : Les étoiles étant marquées en rouge, elles appartiennent à un catalogue temporaire qui sera vidé une fois que le bouton*Annotation* aura été désactivé.

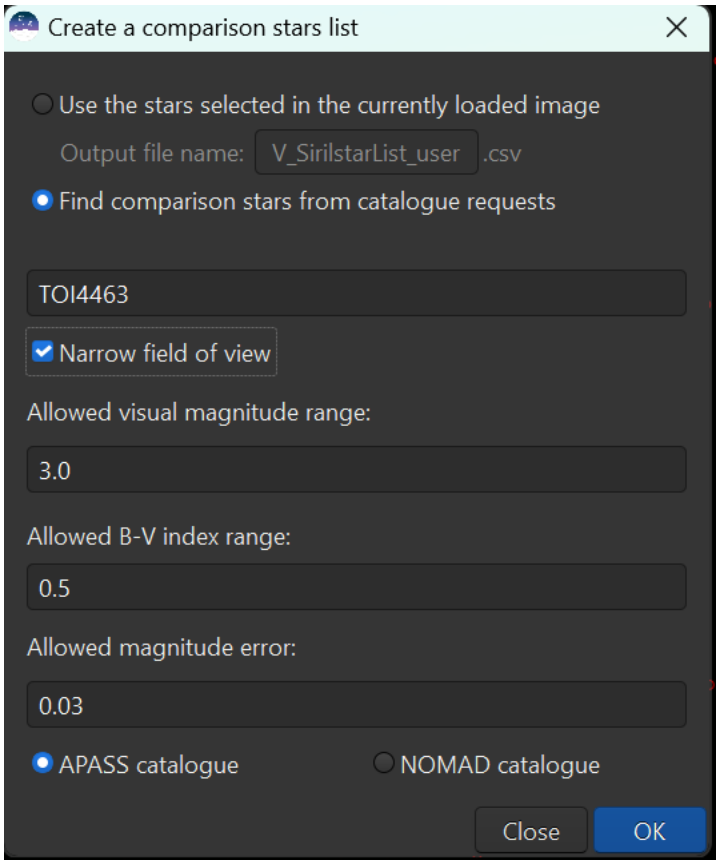
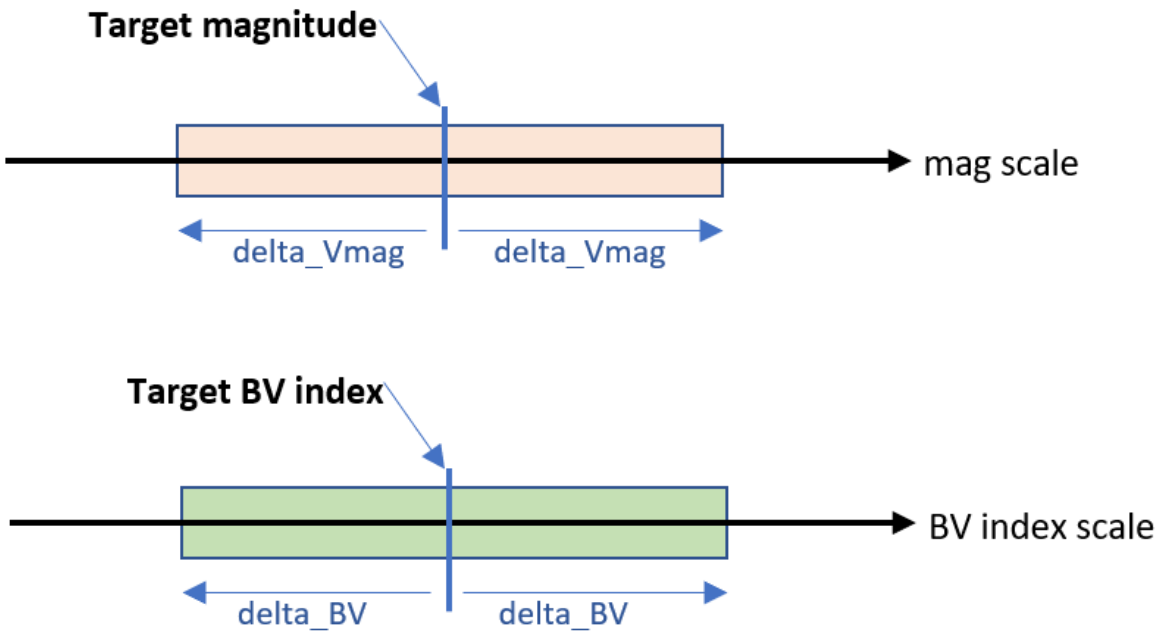


FIG. 18 – Interface graphique de base pour la fenêtre de comparaison des étoiles.



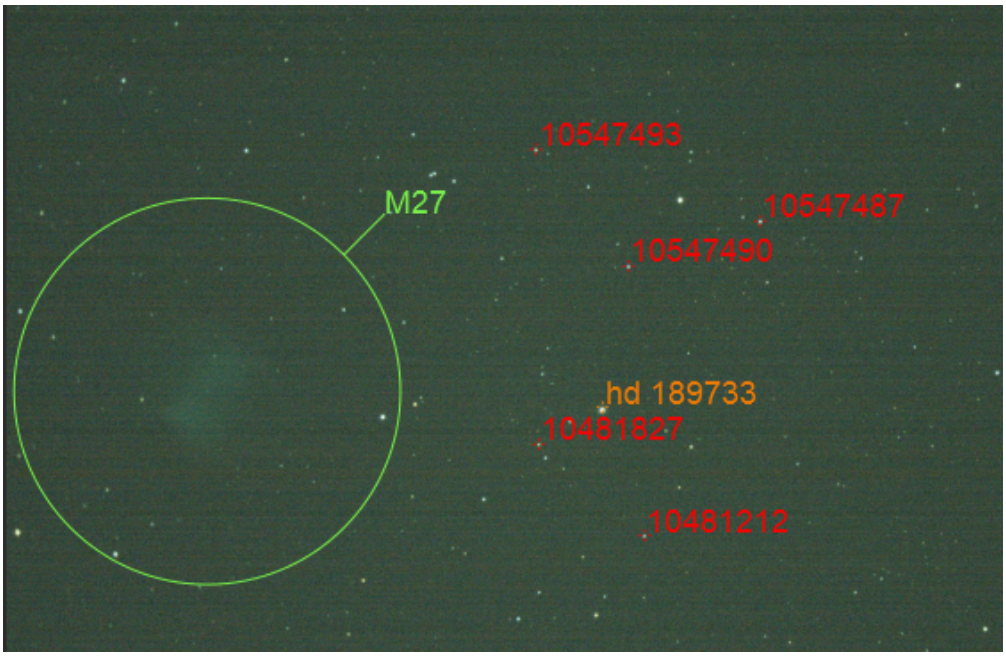


FIG. 19 – Exemple de rendu après une demande APASS.

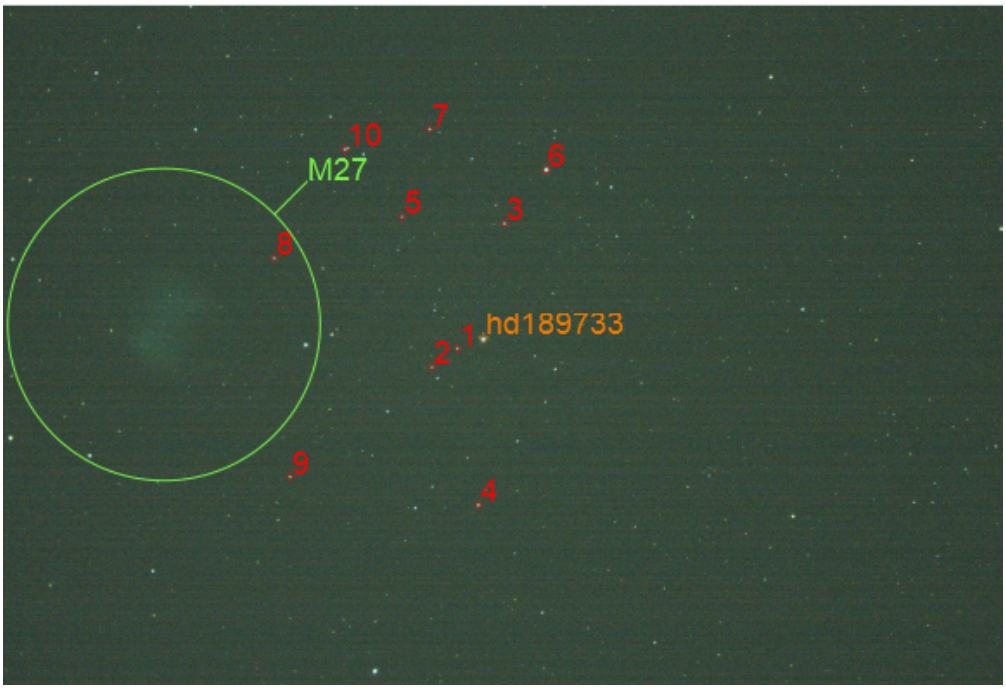


FIG. 20 – Exemple de rendu après une requête NOMAD.

Astuce

Des informations supplémentaires sur le choix des étoiles de comparaison peuvent être trouvées dans le Manuel d'observation des reflex numériques - Version 1.4 page 67 de l'AAVSO.

Liste par commande

Une commande existe également pour effectuer le processus de "recherche d'étoiles de comparaison". Elle vous permet d'affiner davantage votre choix.

Ligne de commande Siril

```
findcompstars star_name [-narrow|-wide] [-catalog={nomad|apass}] [-dvmag=3] [-dbv=0.
↪5] [-emag=0.03] [-out=nina_file.csv]
```

Recherche automatiquement des étoiles de comparaison dans le champ de l'image chargée résolue en astrométrie, pour l'analyse photométrique de la courbe de lumière d'une étoile selon les critères suivants
- the provided name of the star
- the field of view of the image, reduced to a diameter of its height if **-narrow** is passed, avoiding stars in the corners
- the chosen catalog (APASS by default), can be changed with **-catalog={NOMAD|APASS}**
- the difference in visual magnitude from the variable star, in the range [0, 6] with a default of 3, changed with **-dvmag=**
- the difference in color with the variable star, in the range [0.0, 0.7] of their B-V indices with a default of 0.5, changed with **-dbv=**
- the maximum allowed error on Vmag in the range [0.0, 0.1] with a default of 0.03, changed with **-emag=**.

La liste peut éventuellement être sauvegardée sous la forme d'un fichier CSV compatible avec la liste des étoiles de comparaison de NINA, en spécifiant le nom du fichier avec **-out=**. Si le nom fourni est la valeur spéciale **auto**, il est généré en utilisant les paramètres d'entrée

Voir aussi LIGHT_CURVE

Links : *light_curve*

— `-narrow` limite le champ de recherche à un cercle inscrit et centré dans votre image. Cela évite de choisir des étoiles dans une partie potentiellement déformée de l'image.
— Sans l'option `-narrow`, les étoiles de comparaison sont recherchées sur l'ensemble de l'image. Cela peut être utile dans le cas d'un champ de vision peu peuplé.
— `[-catalog={nomad|apass}]` vous permet de choisir entre le catalogue NOMAD et le catalogue APASS (pardéfaut).

Astuce

Les noms affichés seront réduits à de simples nombres, triés par distance croissante du centre de l'image. Ces étiquettes sont uniquement à titre informatif.

— `[-dvmag=3]`, `[-dbv=0.5]` et `[-max_emag=0.03]` représentent respectivement les critères **delta_Vmag** , **delta_BV** et **max_emag** comme expliqué précédemment.

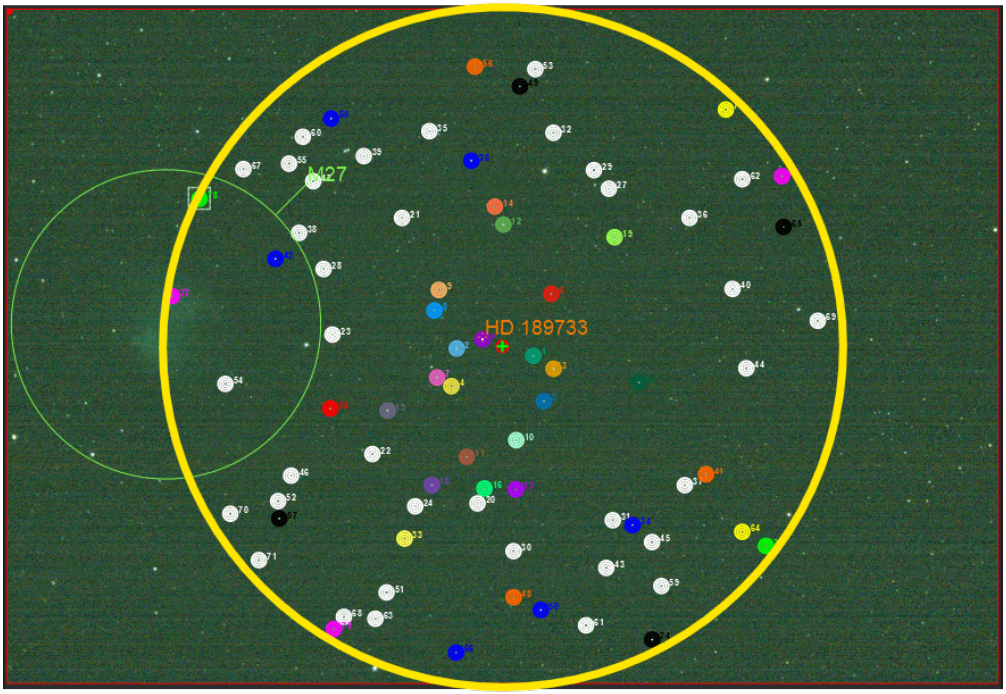


FIG. 21 – Vue du champ de recherche limité `-narrow`. (le cercle jaune n'est là qu'à titre d'illustration)

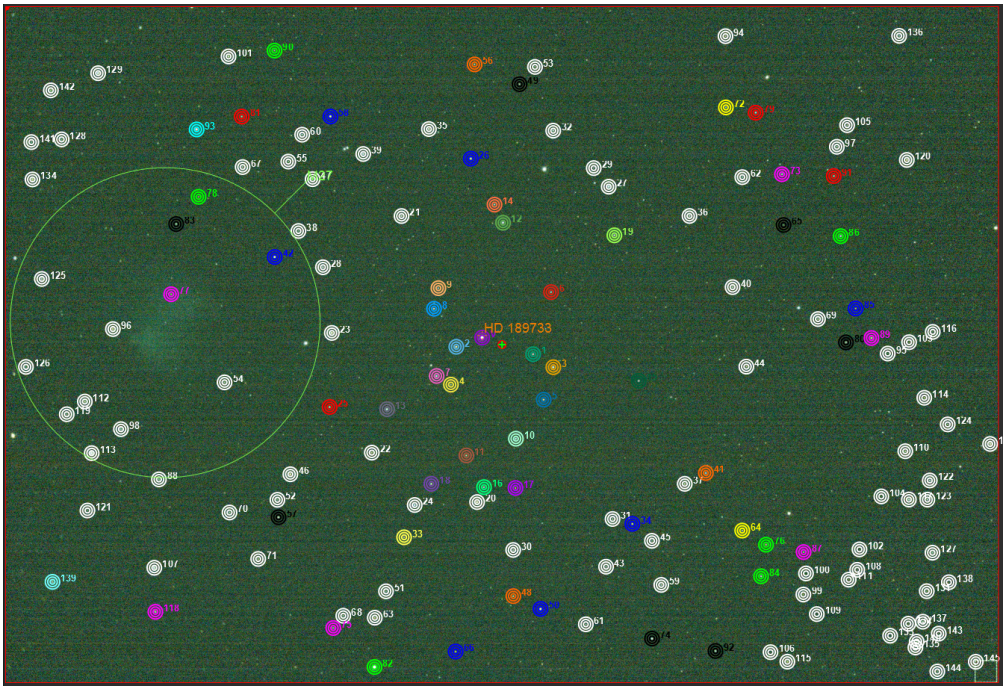


FIG. 22 – Vue du champ de recherche élargi.

— `[-out=nina_file.csv]` définit le nom du fichier CSV.

Astuce

APASS signifie AAVSO Photometric All Sky Survey. Des informations supplémentaires sont disponibles ici.

Les relevés photométriques de l'AAVSO (APASS DR9) fournissent des magnitudes calibrées dans l'intervalle 7,0 < Vmag < 17,0 pour le ciel entier, dans les bandes passantes BVugriZsY.

La photométrie est précise à environ 0,02 mag, et l'astrométrie est précise à environ 0,15 arcsec (voir Utilisation d'APASS et de 2GSS pour l'étude des étoiles variables).

Enlever les étoiles variables potentielles

Parmi les étoiles renvoyées par APASS ou NOMAD, quelques-unes peuvent être identifiées comme des étoiles variables. Si une (ou plusieurs) de ces étoiles sont sélectionnées comme étoiles de comparaison, leur variation de flux peut affecter le résultat final pour l'étoile variable étudiée.

La fonction de liste d'étoiles de comparaison automatique de Siril est donc capable de trouver ces étoiles aberrantes et de les enlever en conséquence.

Trois catalogues principaux peuvent être utilisés pour détecter les étoiles variables :
— Le Catalogue général des étoiles variables (GCVS). Fournit plus de 89 000 étoiles variables.
— Le AAVSO VSX, il liste toutes les étoiles qui ont une entrée dans l'index international des étoiles variables de l'AAVSO.
— Le GAIA varisum , un sous tableau de GAIA DR3 listant les "objets variables potentiels".
Ces 3 catalogues peuvent être utilisés un par un ou tous en même temps pendant le processus de mise à l'écart. Mais vous devez savoir que chaque catalogue sélectionné entraîne une demande en ligne supplémentaire.

En principe, seul le catalogue GAIA Varisum est sélectionné. L'utilisateur peut toutefois faire son propre choix.

Les catalogues utilisés sont gérés par une variable globale `photometry.discard_var_catalogues` selon le tableau suivant :

Tableau 1 – Table pour proprement définir `photometry.discard_var_catalogues`.

Valeur de la variable	Binaire	GCVS	AAVSO VSX	GAIA Varisum
0	0000			
1	0001	✓		
2	0010		✓	
3	0011	✓	✓	
4	0100			✓
5	0101	✓		✓
6	0110		✓	✓
7	0111	✓	✓	✓

(On peut voir dans le tableau précédent que la variable peut être écrite en format binaire, avec `bit#0` comme GCVS, `bit#1` comme VSX et `bit#2` comme GAIA-Varisum).

On peut lire cette variable avec la commande :

```
get photometry.discard_var_catalogues
```

On peut assigner une autre valeur à la variable avec la commande :

```
set photometry.discard_var_catalogues=4
```

Par exemple, avec la commande :

```
set photometry.discard_var_catalogues=7
```

Les 3 catalogues sont vérifiés et la console affiche :

```
Contacting server
-> 5 variable stars found within the image from AAVSO Variable stars
Contacting server
-> 1 variable stars found within the image from GCVS
Contacting server
-> 8 variable stars found within the image from Gaia DR3 Variability
```

Structure du fichier de sortie

En utilisant la commande `findcompstars` et en réglant `-out=auto` ou le bouton GUI *Étoiles de comparaison*, on obtiendra un nom de fichier avec la structure suivante :

`[star_name]_SirilstarList_[delta_Vmag]_[delta_BV]_[max_emag]_[catalogue].csv`

Des informations supplémentaires sont également fournies (sous forme de commentaires) dans l'en-tête du fichier :

```
# Sorted comparison stars for KELT-16 from APASS according to the following criteria
# Siril: 12 stars, dVmag 1.00, dBV 0.30, max e_mag 0.03
type,name,ra,dec,mag
Target,KELT-16,314.268494,31.661009,11.72
Comp1,1,314.256578,31.660406,12.307
Comp1,2,314.317434,31.754368,12.431
Comp1,3,314.305316,31.546814,11.682
Comp1,4,314.263680,31.541927,12.084
Comp1,5,314.465451,31.626255,12.084
Comp1,6,314.459388,31.596876,11.979
Comp1,7,314.452316,31.746360,11.504
Comp1,8,314.437181,31.827260,11.24
Comp1,9,314.314246,31.879192,12.313
Comp1,10,314.431752,31.839763,10.88
Comp1,11,314.006569,31.666214,11.896
Comp1,12,314.304606,31.925116,11.713
```

CHAPITRE 17

Fonction graphique

Siril dispose d'un onglet permettant d'afficher des graphiques des données calculées lors de l'alignement ou d'autres calculs. Cet onglet est très puissant et permet un tri facile des images ainsi qu'une analyse approfondie de celles-ci. Le raccourci pour accéder à cet onglet est `F5`.

17.1 Graphique d'alignement

Afin d'améliorer le tri manuel de vos images alignées, des capacités de traçage ont été ajoutées dans l'onglet Graphique. Après avoir terminé l'alignement de votre séquence (ou lors du chargement d'une séquence alignée), une liste déroulante permet maintenant de sélectionner les paramètres d'intérêt pour le traçage et le tri de vos données.

Vous pouvez aussi choisir de choisir de tracer un paramètre *vs* un autre. Les éléments disponibles dans la liste déroulante sont les suivants :

- **FWHM** : Ceci est la largeur moyenne à mis hauteur, l'un des critères les plus courants pour juger de la qualité des images du ciel profond.
- **Rondeur** : La rondeur r est calculée comme le rapport de $\frac{\text{FWHMy}}{\text{FWHMx}}$.
- **wFWHM** : Il s'agit d'une amélioration d'une simple FWHM. La FWHM est pondérée par le nombre d'étoiles dans l'image. Pour une même mesure de FWHM, une image avec plus d'étoiles aura une meilleure wFWHM qu'une image avec moins d'étoiles. Elle permet d'exclure beaucoup plus d'images parasites en utilisant le nombre d'étoiles détectées par rapport à l'image de référence.
- **Fond de ciel** : Valeur moyenne du fond du ciel.
- **# Étoiles** : Il s'agit du nombre d'étoiles utilisées pour l'alignement.
- **Position X** : Décalage en X par rapport à la référence.
- **Position Y** : Décalage en Y par rapport à la référence.
- **Qualité** : Ce critère est un nombre dans la plage [0, 1] qui définit la qualité des images qui ont été traitées par un algorithme d'alignement planétaire.

Cliquez sur l'un des points de données pour exclure l'image ou pour l'ouvrir. Cette dernière option permet de charger l'image et de faire apparaître le sélecteur d'image. Le paramètre choisi pour les valeurs Y est reflété dans la dernière colonne du sélecteur, qui peut ensuite être utilisé pour trier, revoir et sélectionner ou désélectionner des images de la séquence.

Vous pouvez également sélectionner/désélectionner en masse plusieurs points de données en dessinant une sélection sur le graphique. Les informations en haut de la sélection indiquent le nombre de points sélectionnés, ainsi que les valeurs

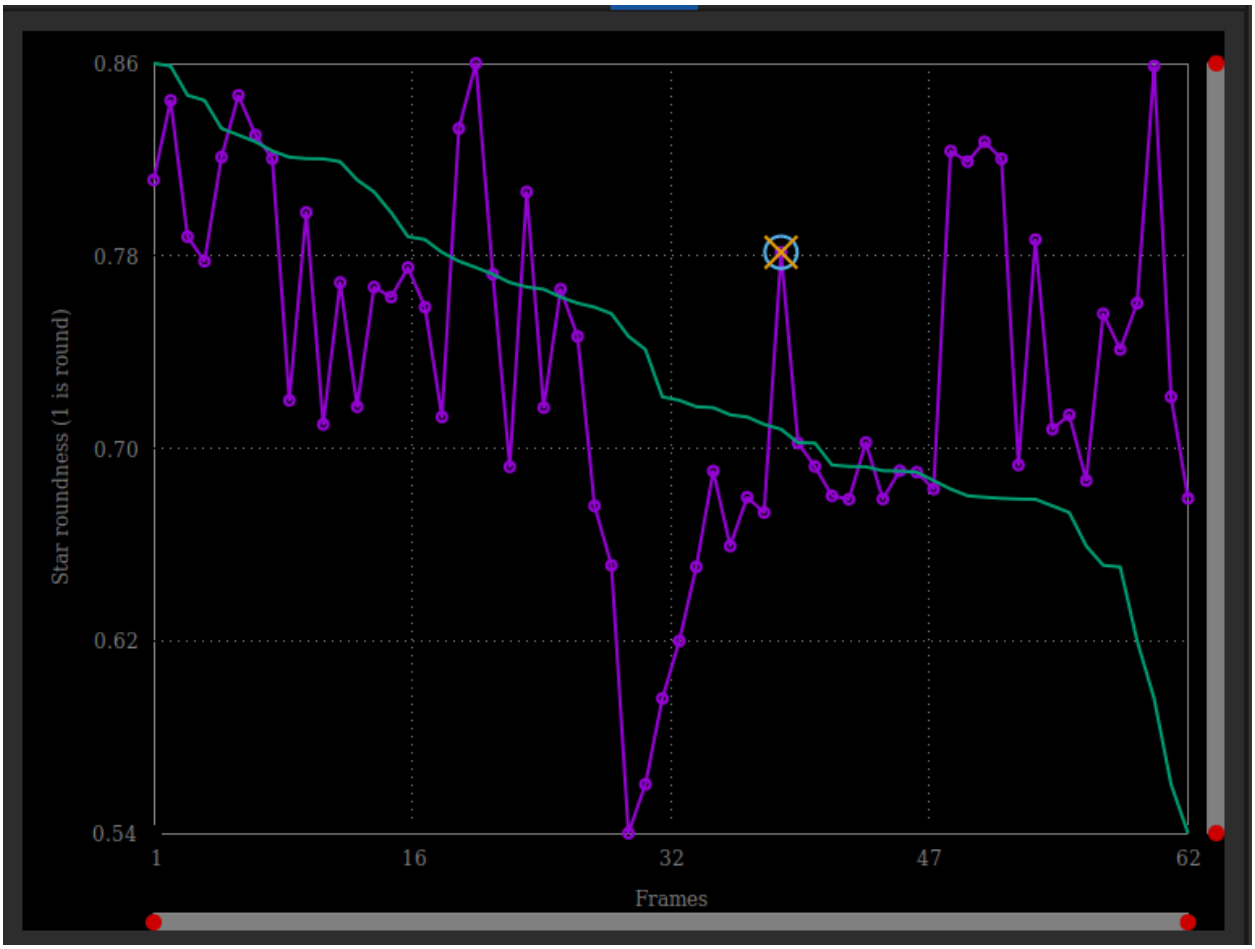


FIG. 1 – Onglet graphique après un alignement global.

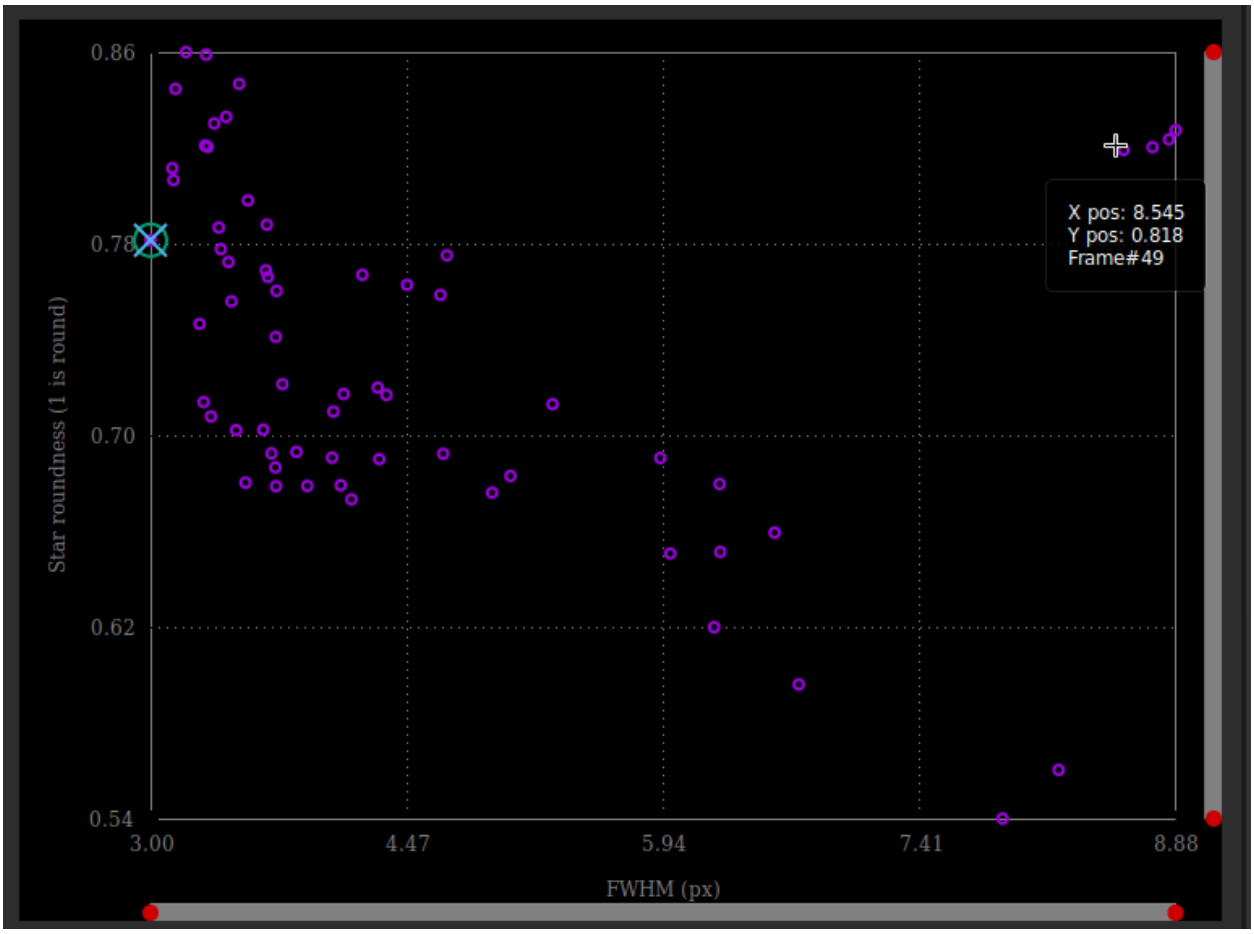


FIG. 2 – Les valeurs de la rondeur en fonction de la FWHM sont affichées sous forme de diagramme de dispersion. Passez la souris sur les différents points de données pour afficher les valeurs X et Y, ainsi que le numéro de l'image correspondante.

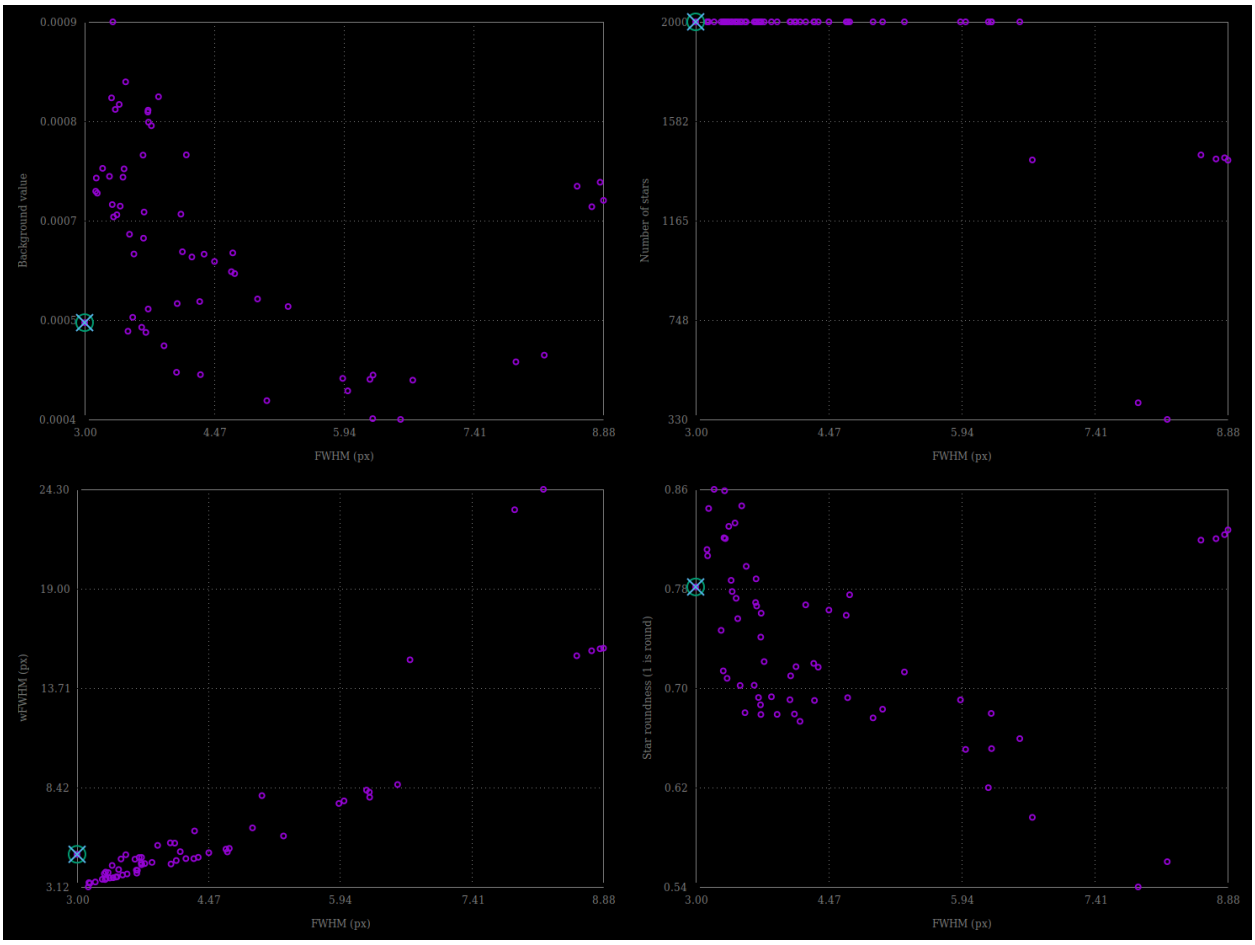


FIG. 3 – Possibilité différente de graphiques réalisés avec le même ensemble d'images.

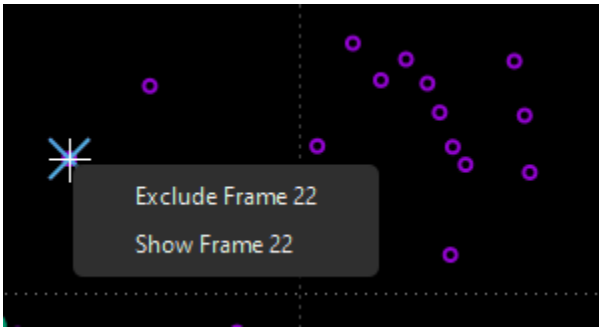


FIG. 4 – Cliquez avec le bouton droit de la souris sur un point de données pour l'exclure ou pour le charger dans l'aperçu de l'image

limites de votre sélection. Vous pouvez remodeler la sélection comme vous le feriez avec une sélection dessinée dans la vue Image. Une fois satisfait de votre sélection, un clic droit affichera un menu permettant de conserver ou d'exclure les points, ou de régler le zoom sur la sélection.

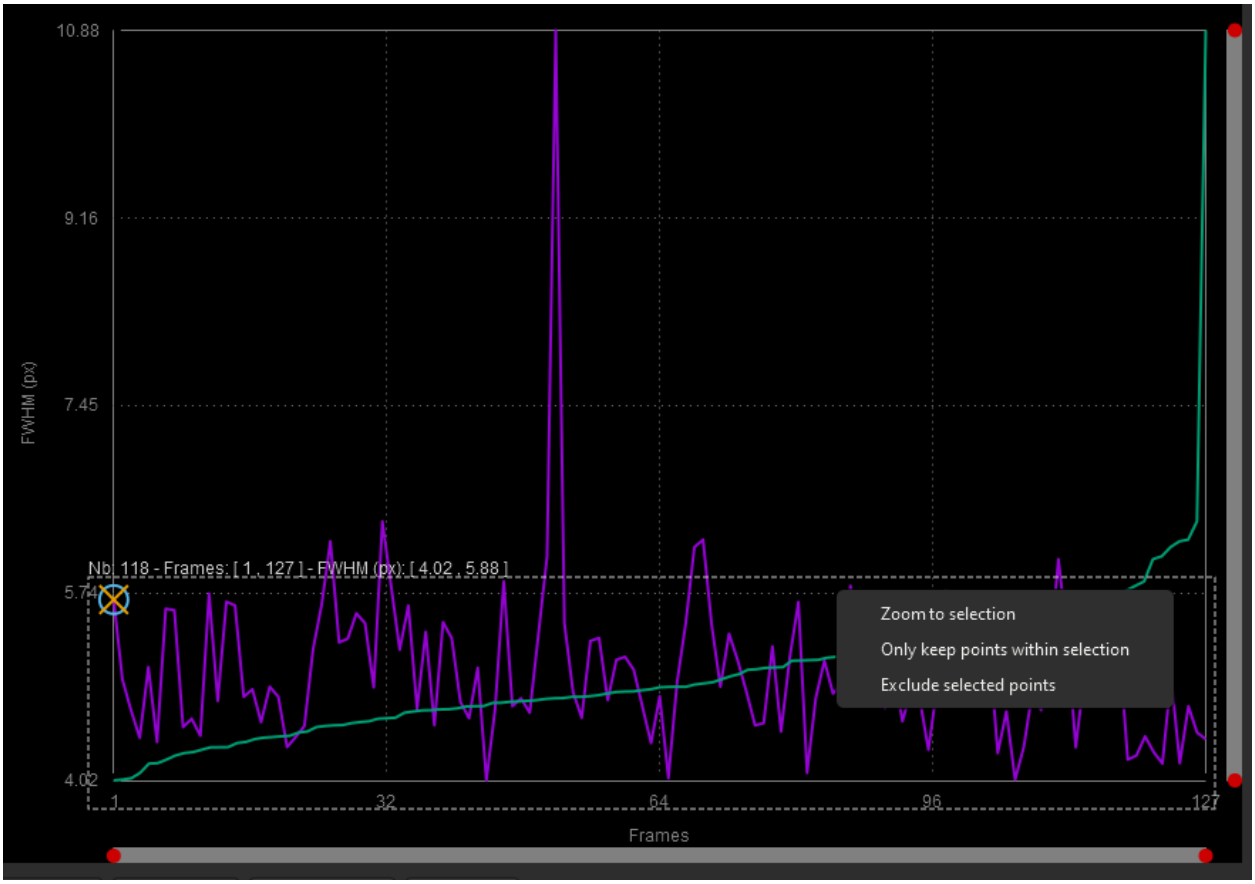


Fig. 5 – Cliquez avec le bouton droit de la souris sur une sélection dessinée, pour la sélectionner/désélectionner en masse ou pour zoomer

17.2 Graphique de photométrie

En complément du tri et du filtrage des images de la séquence, vous pouvez également effectuer une PSF sur une étoile pour la séquence complète. La procédure est détaillée dans la page *photométrie*. Ensuite, l'élément de photométrie de la première liste déroulante devient sensible et est automatiquement sélectionné. L'autre liste déroulante contient les éléments suivants :

- **FWHM** : Largeur maximum à mi-hauteur, tel que défini ci-dessus.
- **Rondeur** : La rondeur r est calculée comme le rapport de $\frac{\text{FWHMy}}{\text{FWHMx}}$.
- **Amplitude** : C'est la valeur maximale de la fonction ajustée, située aux coordonnées du centroïde.
- **Magnitude** : Magnitude relative de l'étoile analysée.
- **Fond de ciel** : Moyenne de la valeur du fond du ciel local prise dans un anneau.
- **Position X** : Décalage en X par rapport à la référence.
- **Position Y** : Décalage en Y par rapport à la référence.
- **RSB** : Un estimateur du rapport signale sur bruit.

En photométrie, contrairement à l'alignement, il n'est pas possible de changer l'axe des X. Et seul le nombre d'images peut être utilisé (ou le jour julien).

- Le bouton *Effacer les courbes* combine les menus *Tout effacer* et *Effacer la dernière* pour effacer soit toutes les courbes photométriques, soit la dernière tracée.

17.2.1 Menu Outils

Depuis Siril 1.4, de nombreuses options ont été déplacées dans le menu *Outils*, situé à droite du menu *Traitement de l'image*. Ce menu sera détaillé dans la section *Photométrie*.

17.3 Options partagées

Plusieurs fichiers de sortie peuvent être produits en cliquant sur le bouton *Production* :

- *Afficher les données valides dans Siril Plot* est une option qui affiche tout ce qui se trouve dans l'onglet graphe dans une nouvelle instance de Siril Plot.
- Le bouton *Exporter en CSV* exporte le graphique affiché dans un fichier CSV.

Le bouton *Affichage* permet de changer les unités de l'axe. Il contient, en fonction de ce qui est disponible, plusieurs boutons dans un menu déroulant :

- Si l'échantillonnage des images est connu, le bouton *arcsec* affiche la FWHM en arcsec au lieu du pixel.
- Dans le cas de la photométrie uniquement (donc pas vraiment dans les options partagées), il est également possible de choisir *Date julienne* comme unité de l'axe des x.

Astuce

Survolez le mot "Survoler pour la légende" pour afficher la légende du graphique.

- Courbe violette : tracé réel tel que configuré avec les sélecteurs X et Y.
- Courbe verte : valeurs triées par ordre de qualité décroissante.
- Cercle : la valeur de l'image de référence.
- Croix : la valeur de l'image actuellement chargée.

17.4 Interaction avec le graphique

Voici un résumé des interactions possibles avec la fenêtre du graphique :

- **Clic gauche** dans un curseur : met le point rouge le plus proche sur celui-ci
- **Double-clic** sur un curseur : réinitialise cet axe
- **Clic droit + glisser** sur un curseur : modifier le zoom de cet axe
- **Clic gauche + glisser** dans la zone graphique : dessine une sélection
- **Clic gauche + glisser** sur un bord de la sélection : redimensionner la sélection
- **Double-clic** dans la zone de graphique : réinitialise le zoom des 2 axes
- **Clic droit** lorsqu'une sélection est active : afficher le menu pour : zoomer sur la sélection/ garder seulement les points de la sélection/ exclure les points de la sélection
- **Clic gauche** quand une sélection est active : supprimer la sélection

CHAPITRE 18

Statistiques

Il s'agit d'une documentation pour les statistiques de Siril, données par l'interface graphique (GUI) à partir du menu *Outils →Analyse de l'image* puis sélection de *Statistiques...* ou en utilisant la commande stat. Notez que lorsque vous utilisez l'interface graphique, il est possible de dessiner une sélection dans l'image chargée et que, ce faisant, les statistiques sont calculées sur les pixels de la région.

L'option *Par canal CFA* permet de calculer les statistiques pour chaque canal R, V, B d'une image CFA, même si l'image n'a pas été dématricée.

Astuce

When a CFA image is loaded, the selection will be constrained to a minimum size of 2x2 for Bayer images and 3x3 for X-Trans images so that there is at least one pixel in each Bayer channel within the selection, to ensure that it is possible to generate statistics.

Plusieurs de ces valeurs sont des mesures de la dispersion statistique.

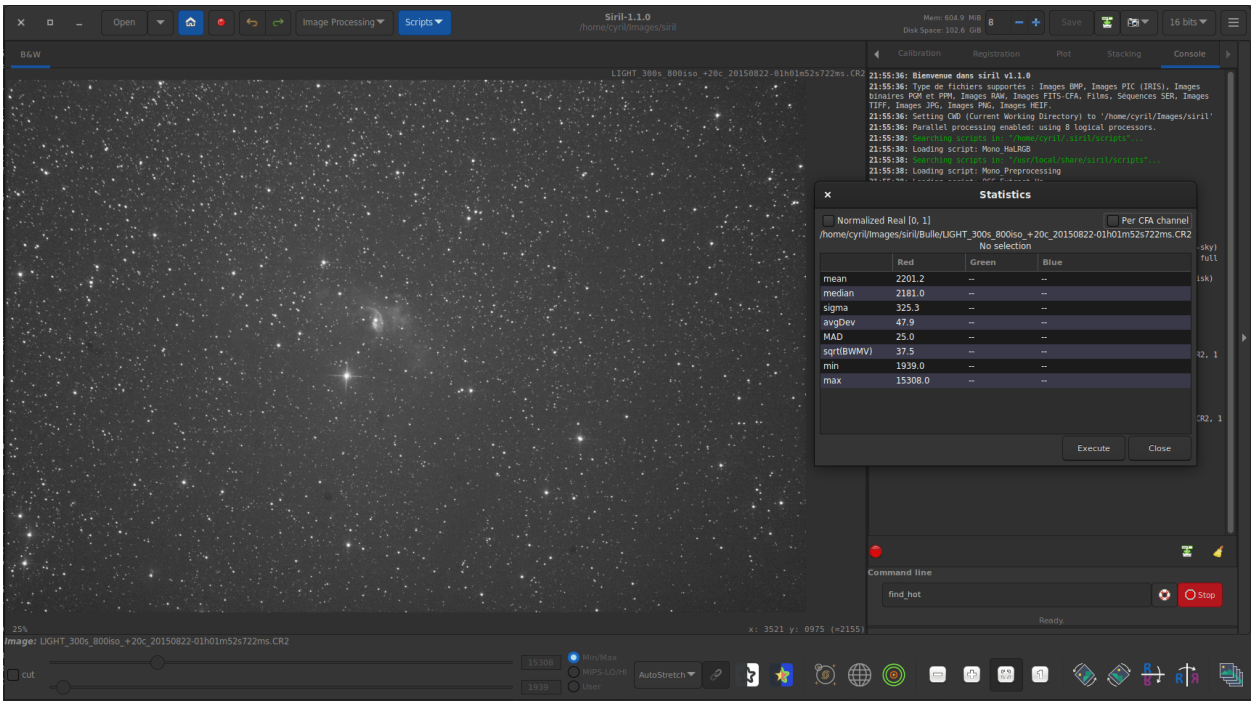


FIG. 1 – Statistiques un canal pour une image CFA. Les valeurs données ne sont pas vraiment pertinentes dans ce cas.

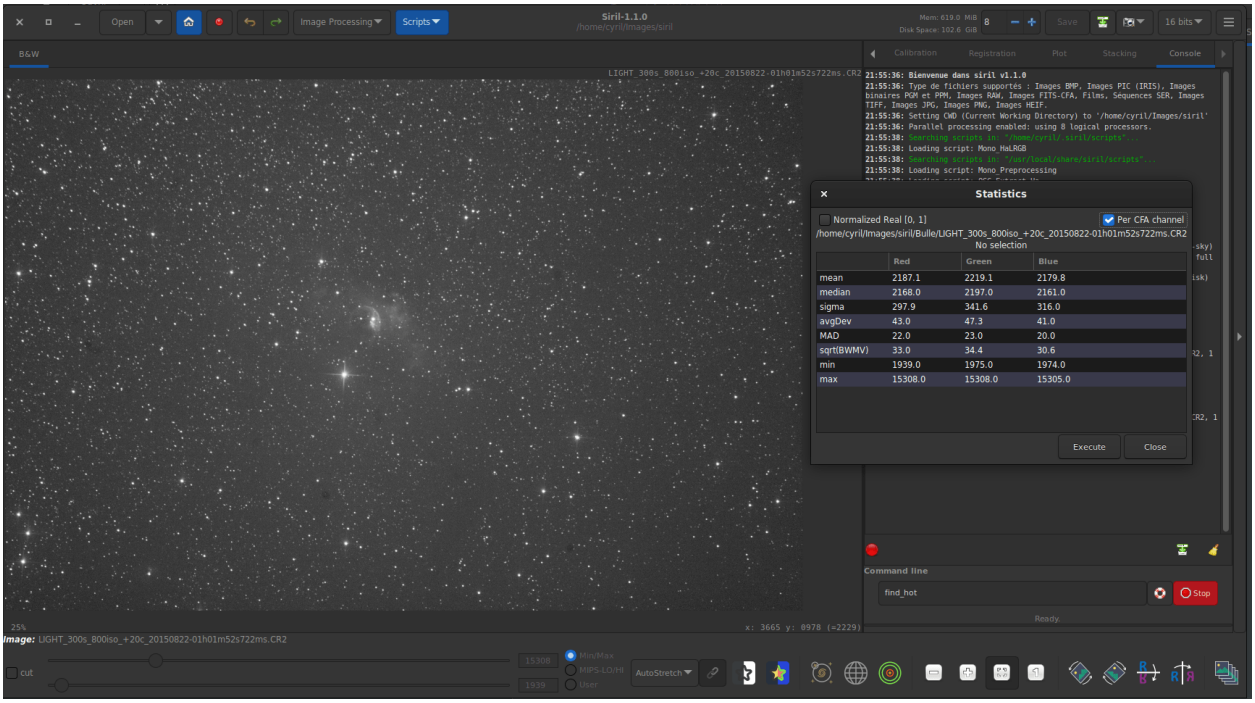


FIG. 2 – Statistiques trois canal pour une image CFA.

Ligne de commande Siril

```
stat [-cfa] [main]
```

Renvoie les statistiques de l'image courante, la liste de base par défaut ou la liste principale si **main** est passé. Si une sélection est faite, les statistiques sont calculées dans la sélection. Si **-cfa** est passé et que l'image est CFA, les statistiques sont faites sur les extractions par filtre

18.1 Estimateurs

18.1.1 Moyenne

C'est la moyenne arithmétique, également appelée moyenne. Elle est calculée en faisant la somme des valeurs des pixels, divisée par le nombre de pixels d'un canal d'image.

18.1.2 Médiane

La médiane est la valeur qui sépare la moitié supérieure de la moitié inférieure d'un ensemble de données. En général, elle représente la valeur de l'arrière-plan d'une image astronomique.

18.1.3 Sigma

Également connue sous le nom d'écart-type, notée σ, c'est une mesure de la dispersion des pixels de l'image basée sur les différences au carré par rapport à la moyenne. La valeur sigma d'une sous-image contenant uniquement l'arrière-plan représentera le bruit de l'image.

18.1.4 Bruit de fond

Cet estimateur est disponible dans l'interface graphique à partir du menu *Outils* → *Analyse de l'image* → *Estimation du bruit* et il est également affiché à la fin de l'empilement.

Il s'agit d'une mesure du niveau de bruit estimé dans l'arrière-plan de l'image, pour les pixels ayant une valeur suffisamment faible pour être considérés comme de l'arrière-plan. C'est un processus itératif basé sur k.sigma (un facteur de l'écart-type au-dessus de la médiane), il n'y a donc pas de seuil fixe pour déterminer ce qui est suffisamment bas.

Ligne de commande Siril

```
bgnoise
```

Renvoie le niveau du bruit de fond de l'image chargée en mémoire

18.1.5 avgDev

La déviation moyenne, également appelée AAD pour average absolute deviation ou mean absolute deviation. Pour comprendre ce qu'est la déviation moyenne, il faut comprendre ce qu'est la déviation absolue. La déviation absolue est la distance entre chaque valeur de l'ensemble de données et la moyenne de cet ensemble (dans ce cas-ci) ou la médiane (pour MAD ci-dessous). En prenant toutes ces déviations absolues, en trouvant la moyenne, la déviation moyenne absolue est calculée. Pour simplifier, si l'écart type est la déviation quadratique moyenne par rapport à la moyenne, la déviation moyenne est sa version linéaire.

18.1.6 MAD

L'écart absolu médian est une mesure robuste de la dispersion d'un ensemble de données. L'écart absolu et l'écart type sont également des mesures de dispersion, mais ils sont davantage affectés par des valeurs extrêmement élevées ou extrêmement faibles. Il est similaire à l'écart moyen ci-dessus, mais il est relatif à la médiane au lieu de la moyenne.

18.1.7 BWMV

Le biweight midvariance est un autre outil pour mesurer la dispersion d'un ensemble de données, encore plus robuste que les autres cités ci-dessus aux valeurs aberrantes. Il écarte les points de données trop éloignés de la médiane et calcule une variance pondérée, les poids diminuant au fur et à mesure que les points de données s'éloignent de la médiane. L'estimateur de la dispersion est la racine carrée (marquée comme $\sqrt{BWMV}$) de cette valeur.

18.1.8 Position et échelle

Ces paramètres, souvent appelés familièrement échelle et décalage, ne sont pas affichés dans les interfaces utilisateur mais sont calculés en interne par Siril. Afin d'aligner les histogrammes des différentes images pour les normaliser avant de les empiler, il faut calculer où ils se situent en termes de niveau et quelle est leur étendue en termes de dispersion. Un estimateur valable de l'emplacement pourrait être la médiane, tandis que le MAD ou le $\sqrt{BWMV}$ pourrait être utilisé pour l'échelle. Toutefois, afin d'accroître la robustesse des mesures, les pixels situés à plus de $6 \times$ MAD de la médiane sont éliminés. Sur cet ensemble de données découpé, la médiane et $\sqrt{BWMV}$ sont recalculées et utilisées respectivement comme estimateurs d'emplacement et d'échelle. Ils sont calculés par rapport à l'image de référence d'une séquence dans Siril.

CHAPITRE 19

Siril pour les scientifiques

Depuis des années Siril est un outils puissant pour les astronomes amateurs. Avec la possibilité de produire de courbes de lumière, rechercher dans des catalogues en lignes, réaliser des opérations sur une séquence d'images, et bien plus depuis les récent ajouts, il est devenu un bon outil pour de l'analyse scientifique. Cette page est un guide rapide pour les scientifiques qui veulent en apprendre plus rapidement sur Siril, sous forme d'un FAQ.

19.1 Comment Siril peut-il m'aider dans mes recherches ?

Siril's default file format is FITS files, containing either a single image or several (see the FITS cubes question below). Siril has always been developed with data accuracy in mind. Displaying an image can be done in various ways without destroying input data.

Siril has many tools that can help scientific observations : star detection, elaborate plate solving, registration of an image sequence using stars or using astrometric solutions, aperture photometry with magnitude calibration using catalogues, automated light curve creation, object lookup from online catalogues (stars, galaxies, solar system objects). Many of those operations can be done from the GUI or from text inputs (commands or scripts), some operations produce comma-separated values (CSV) files for results, some can produce graphs directly in Siril.

19.2 Quelle version de Siril dois-je installer ?

Siril follows an even/odd versioning system, e.g. 1.2.x was a stable series whereas 1.3.x was a development series. For scientific use it is recommended to install the latest version of the latest stable series : download links can be found at siril.org. Beta versions, those distributed on the website, can also be considered quite stable for general use.

19.3 Comment puis-je contacter les développeurs ?

The official forum is on pixls.us. If you want to contact us privately you can send us a direct message there or on gitlab. Our email addresses can also be found in the AUTHORS file. If you speak French you can find us on Astrosurf or Webastro. Feel free to reach out if you have specific needs for your research.

19.4 Quelle est la précision des résultats de Siril ?

Many algorithms in Siril rely on peer-reviewed published algorithms which can be found in the documentation page of the features. We are testing Siril often, sometimes comparing results with other tools, but we don't have the resources to have a automated tests on all pieces of the software, so data comes with no warranty. We know some things could be improved, for example error computation on magnitudes, which depend on camera gain, not always available in the FITS header.

Les données d'astrométrie sont aussi fiable que les autres et prend en charge la convention WCS FITS.

19.5 Comment citer Siril ?

Si vous utilisez Siril dans vos recherches, veuillez citer :

> Richard, C., et al. (2024). Siril : An Advanced Tool for Astronomical Image Processing. *Journal of Open Source Software*, 9(102), 7242. https://doi.org/10.21105/joss.07242

Format BibTeX :

```
@article{Richard2024,
  title={Siril: An Advanced Tool for Astronomical Image Processing},
  author={Richard, Cyril and others},
  journal={Journal of Open Source Software},
  volume={9},
  number={102},
  pages={7242},
  year={2024},
  publisher={The Open Journal},
  doi={10.21105/joss.07242}
}
```

19.6 Comment puis-je ouvrir des cubes FITS avec Siril ?

Les cubes FITS sont disponibles en deux formes pour Siril :

- ceux qui contiennent plusieurs images de même taille et de même format sont appelés séquences FITS, ou fitseq en abrégé, et servent de format alternatif pour représenter les séquences d'images ;
- those that contain several images of changing properties, or even tables and other data formats. Siril will only read image data, but is able to work on each image of such FITS cubes if an option is enabled in settings (named *Allow FITS cubes to have images of different sizes* in *FITS Options*). Most sequence operations will not be available on this kind of file.

Siril permet également d'extraire des images à partir de cubes FITS pour les enregistrer dans des fichiers FITS individuels ou dans d'autres formats si nécessaire.

19.7 Comment mesurer la FWHM dans une image ?

Siril peut vous donner la FWHM d'une seule étoile si vous tracez une sélection autour de celle-ci dans l'interface graphique, puis cliquez avec le bouton droit et sélectionnez PSF, ou pour les étoiles dans une zone de l'image ou l'image entière. Vous pouvez également utiliser le bouton *quick PSF* dans la barre d'outils principale.

More generally : open your image and open the *Dynamic PSF tool* (from the *Tools* menu). From there clicking on the first button will start the star detection, within the selected area if you drew one. Clicking on the Sigma icon will give mean values for all detected stars. See also the next question.

19.8 Comment utiliser la détection des étoiles (extraction des sources) ?

Siril's source extractor can be found in the GUI in the *Dynamic PSF tool*, it is also available as the *findstar* command. It is normally only able to detect stars and will struggle if two stars are too close from each other, or if stars have asymmetrical or very non-Gaussian shapes, but there are many settings that you could adjust to your needs, including brightness, amplitude, roundness thresholds or star shape (Gaussian or Moffat).

The GUI will display the found stars by clicking on the button on the left of the lower bar, giving quick feedback on the detection settings. The GUI and the command can produce a CSV file that can then be used to check extracted sources in an image or in a sequence of images (command only).

La liste des étoiles peut être triée par propriété, en cliquant sur une ligne l'étoile correspondante sera mise en surbrillance sur l'image.

19.9 Quels sont les outils d'astrométrie disponibles ?

Siril has now a very high quality astrometric solver (plate solver) and tools that interact with the astrometric solution. All is based on the FITS WCS convention and on WCSLIB. Here is a list of tools related to astrometry :

- Plate solving with local Gaia DR3 catalogue extract or with remote full Gaia DR3, PPMXL or APASS.
- Display stars from the catalogue on image, to have a visual feedback of the accuracy of the astrometric solution, using the *conesearch* command.
- *Query* catalogues en ligne pour un objet à partir de son nom et de la date de l'image.
- Obtenir les coordonnées équatoriales J2000 du centroïde de tout objet détecté.
- Show a mark on image for user provided J2000 coordinates (*show* command).
- Align images on the celestial grid, possibly creating a mosaic (an image bigger than the original).
- Correct for geometrical aberration of images, using a master correction image based on astrometry.

19.10 Peut-il traiter les objets du système solaire ?

Siril can query IMCCE services to find a solar system object coordinates and display the expected position on images. It can also display the known solar system objects in an image. Observer location will be required for those uses and can be configured with MPC codes in the settings. See the *annotations* documentation page.

Soon, Siril will be able to do synthetic tracking of known solar system objects, making them appear on stacking if they are too faint to be seen on single exposures.

But Siril is not made to process closeup view of planets with multi-point stacking like other tools such as PlanetarySystemStacker (PSS) does.

19.11 Siril peut-il fonctionner avec des spectre ?

Siril was not designed to handle spectrometry data. An *Profil d'intensité* tool that shows the graph of pixel intensities can be used for preview of a spectrum. FITS tables are not displayed in any way.

19.12 Siril peut-il remplacer IRAF ?

Certaines fonctionnalités seront absentes, mais beaucoup seront beaucoup plus faciles à utiliser avec Siril. Nous n'avons pas encore de liste.

Siril est développé depuis 2012 et nous n'avons pas l'intention d'arrêter de sitôt. Il compte 3 développeurs actifs et des milliers d'utilisateurs amateurs.

CHAPITRE 20

Scripts

Cette section présente les différentes méthodes de script et d'automatisation disponibles dans Siril.

Astuce

Python scripting was introduced in the 1.3.5 development version. It is currently marked as **EXPERIMENTAL**. This doesn't mean it will eat your data : the interface itself is robust and has been tested through development : the present experimental nature of it is more to do with the fact that we don't yet know what users will do with this new capability and whether there may be issues or limitations that we have not foreseen, perhaps due to the constraints of packaging or consistency across the different operating systems.

Please try it out, either as a user by using scripts published on the scripts repository, or as a script writer. We welcome all your feedback and will aim to refine the interface throughout the 1.4 stable series and 1.5 development series.

If you want to debug your Python scripts at runtime, tick the box next to *Enable Python debug mode*. A tutorial detailing the steps to attach to the Python process is shown in Siril tutorial page.

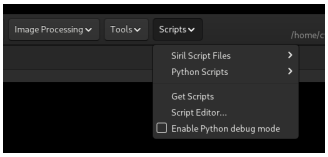


FIG. 1 – Menu script

20.1 Fichiers de Scripts Siril

Siril Script Files are the original format for scripting Siril and remain a simple and useful way of automating a fixed set of tasks.

20.1.1 Scripts Siril

Siril scripts 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.

> **Astuce**
>
> 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.

Les commandes peuvent être saisies dans la ligne de commande située en bas de la fenêtre principale de Siril. Une autre méthode consiste à placer les commandes dans un fichier et à l'exécuter en tant que script. Pour exécuter le fichier à partir de l'interface graphique, ajoutez-le aux répertoires de scripts configurés ou à partir de l'interface graphique, utilisez le symbole `@` de la ligne de commande comme suit :

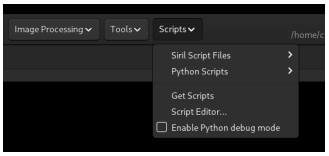
```
@file_name
```

Certaines commandes (*calibrate*,*stack*, et toutes les commandes de sauvegarde) peuvent utiliser des noms de fichiers contenant des variables provenant de l'en-tête FITS. Le format de l'expression est expliqué en détail*ici* et peut être testé en utilisant la commande*parse*.

20.1.2 Utilisation des scripts

Il y a quatre façons d'exécuter un script :

- à partir de l'interface utilisateur graphique, en utilisant le mot clé `@` sur la ligne de commande, suivi du nom du script dans le répertoire de travail actuel,
- depuis l'interface graphique utilisateur, utilisez le menu *Scripts*,



- from the graphical user interface, using the *Script Editor* dialog which can be used to write, save and execute your own scripts or to open and edit an existing script file,
- depuis l'interface de ligne de commande (**`siril-cli`** exécutable), en utilisant l'argument `-s` suivi du chemin du script (voir la page man pour plus d'informations).

When a script is running, the user interface becomes non-interactive with the exception of the Stop button.

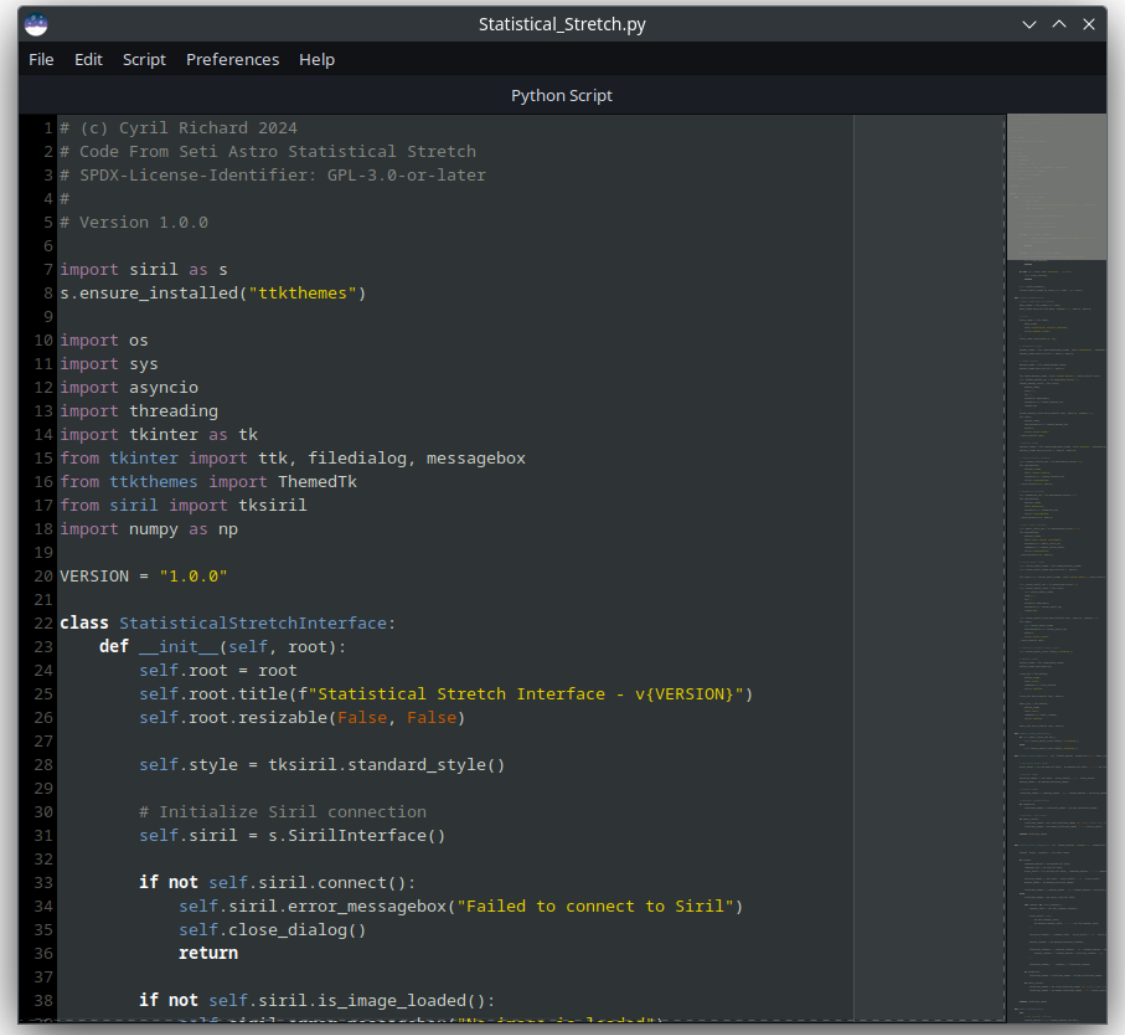
20.1.3 Remplir la liste des scripts

Par défaut, lors de l'installation de Siril, un certain nombre de scripts sont automatiquement installés. Ces scripts intégrés, les scripts officiels, sont développés par l'équipe de développement et leur fonctionnement est garanti : ils sont destinés à couvrir des cas d'utilisation spécifiques.

Ajouter des dossiers de scripts personnalisés

You can, of course, *write your own* and tell Siril where to find them :

- Cliquez sur l'icône *Burger* puis sur *Préférences* (ou appuyer sur `Ctrl+P`).
- Cliquer sur la section *Scripts*.
- Copiez sur une nouvelle ligne le chemin d'accès à l'emplacement où vous les stockerez (créez un dossier sur votre ordinateur ou indiquez un dossier existant).


Statistical_Stretch.py
File Edit Script Preferences Help
Python Script
(c) Cyril Richard 2024
Code From Seti Astro Statistical Stretch
SPDX-License-Identifier: GPL-3.0-or-later
#
# Version 1.0.0
import siril as s
s.ensure_installed("ttkthemes")
import os
import sys
import asyncio
import threading
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
from ttkthemes import ThemedTk
from siril import tksiril
import numpy as np
VERSION = "1.0.0"
class StatisticalStretchInterface:
def __init__(self, root):
self.root = root
self.root.title(f"Statistical Stretch Interface - v{VERSION}")
self.root.resizable(False, False)
self.style = tksiril.standard_style()
# Initialize Siril connection
self.siril = s.SirilInterface()
if not self.siril.connect():
self.siril.error_messagebox("Failed to connect to Siril")
self.close_dialog()
return
if not self.siril.is_image_loaded():

— Cliquez sur l'icône *Rafraichir* juste en dessous.
— Cliquer sur *Appliquer*.

Vous pouvez avoir autant de dossiers définis par l'utilisateur que vous le souhaitez, il vous suffit juste de les ajouter à la liste.

Si vous venez d'ajouter un nouveau script dans l'un des dossiers et que vous souhaitez rafraîchir le menu, tapez la commande*reloadscripts* en ligne de commande ou ouvrez la section*Preferences → Scripts* et utilisez l'icône*Rafraîchir*. Cela permet d'analyser tous les dossiers de la liste et de trouver tous les fichiers portant l'extension *.ssf.

Avertissement

Il est fortement déconseillé de stocker vos scripts personnalisés dans le même dossier que les scripts Siril intégrés. Sous Windows, ils peuvent être effacés lors de l'installation d'une nouvelle version ou empêcher une désinstallation correcte. Sous MacOS, cela casserait le bundle et empêcherait l'utilisation de Siril.

Ne vous inquiétez pas, la liste des emplacements des scripts étant stockée dans votre fichier de configuration, vous devriez les retrouver lors de l'installation d'une version plus récente.

Ajouter des scripts à partir du dépôt Git

Siril supports a git repository at https://gitlab.com/free-astro/siril-scripts This is set to auto update at startup by default, so you will always have access to the latest scripts (auto update can be disabled in preferences). To add scripts from the repository to the Scripts menu, pick the ones you want from the list available in *Preferences → Scripts* or via the *Scripts -> Get Scripts* menu entry. See below for full details of the git repository.

Exécuter des fichiers de script

Script files can also be run directly from the hard disk using the *Run Script Files...* script menu entry. The filechooser shown by this menu defaults to showing recently used scripts, but you can navigate to choose scripts from anywhere accessible on the filesystem.

Résolution de problèmes

For different reasons, it is possible that the *Scripts* menu is empty. This means that the scripts have not been found. If this is the case, please use the following procedure.

— Cliquez sur l'icône *Burger* puis sur *Préférences*.
— Cliquer sur la section *Scripts*.
— Supprimer toutes les lignes dans le champ *Répertoire des scripts* tel que montré dans l'illustration si dessous.
— Si vous utilisez le dépôt de scripts, décochez la case *Récupérer et mettre à jour les scripts...*, puis décochez-la à nouveau et sélectionnez les scripts que vous voulez utiliser.
— Cliquer sur *Appliquer*.
— Fermer et redémarrer Siril.

20.1.4 Scripts intégrés

Tous les scripts intégrés doivent respecter cette structure de fichier :

— **Mono_Preprocessing.ssf** : script pour le prétraitement des appareils photos numériques ou caméras astro monochromes, utilise les biais, les flats et les darks, aligne et empile les images. Pour l'utiliser : mettez vos fichiers (RAW ou FITs) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script.
— **OSC_Preprocessing.ssf** : même script que ci-dessus mais pour les appareils photos numériques ou caméras astro couleur. Pour l'utiliser : mettez vos fichiers (RAW ou FITS) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script.
— **OSC_Preprocessing_BayerDrizzle.ssf** : même script que ci-dessus mais utilisant Bayer Drizzle pour récupérer les couleurs. Pour l'utiliser : placez vos fichiers (RAW ou FITS) dans les dossiers nommés `lights`, `darks`,

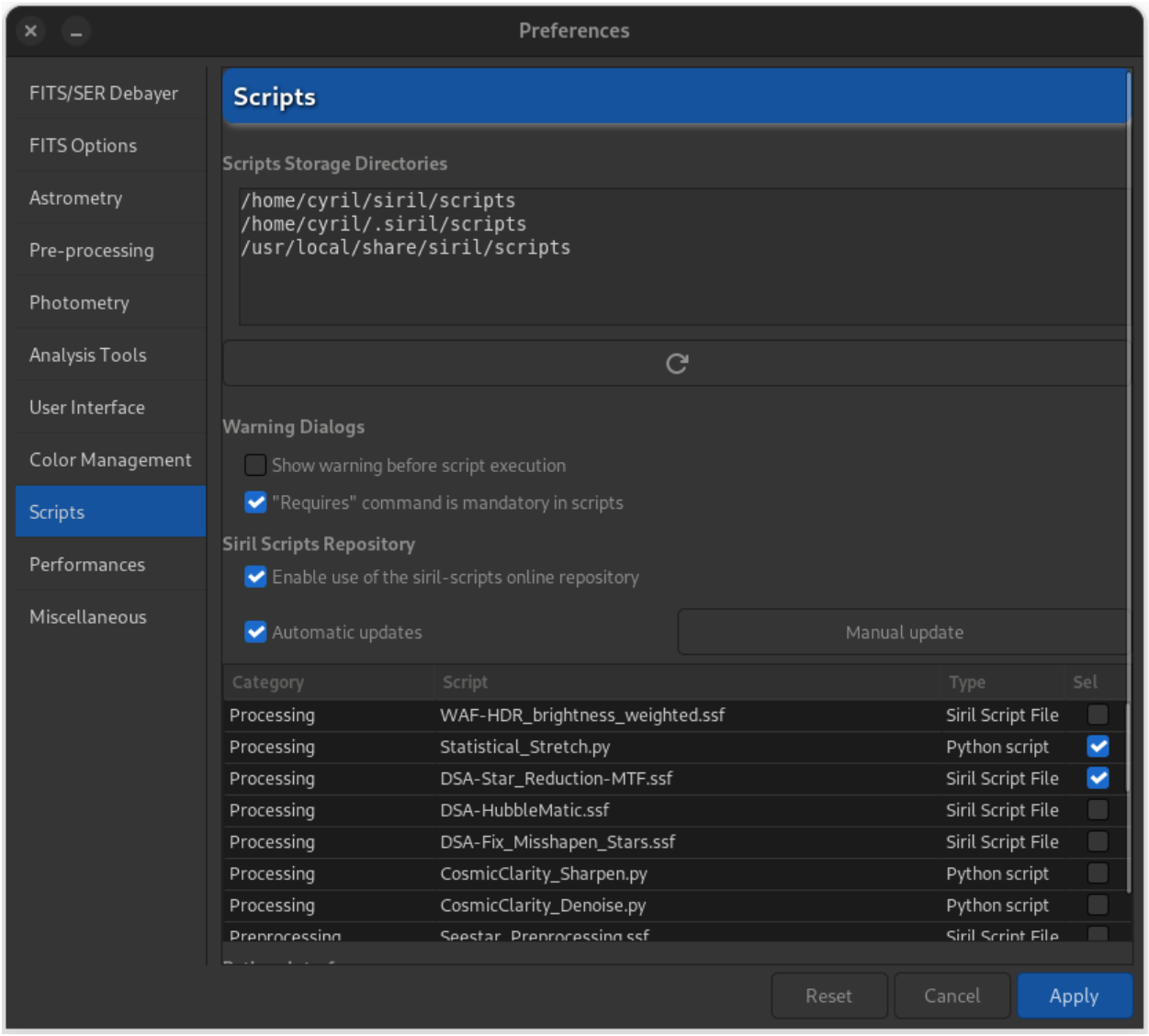
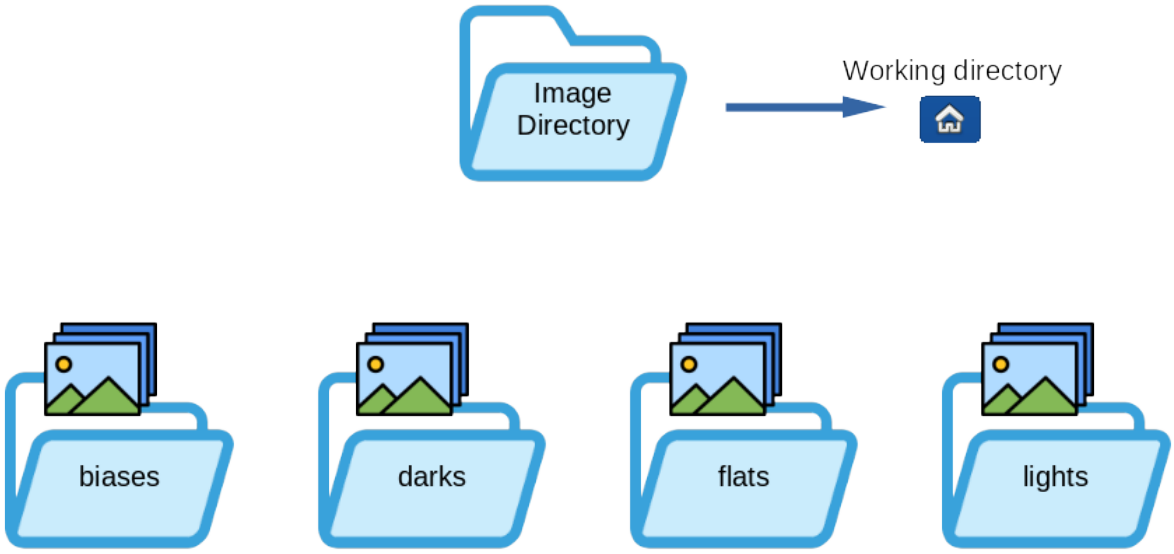


FIG. 2 – Page de préférence des scripts. Les scripts sont chargé depuis les répertoires listé dans *Répertoire des scripts*.



`flats` et `biases` (dans le dossier de travail par défaut de Siril), puis exécutez le script. Il est fortement recommandé de disposer d'une grande quantité de données pour profiter des avantages du Bayer Drizzle et éviter les artefacts disgracieux.

— **OSC_Extract_Ha.ssf** : script pour le prétraitement des appareils photos numériques ou caméras astro couleur, à utiliser avec le filtre Ha ou le filtre à double bande. Ce script extrait la couche Ha de l'image couleur. Pour l'utiliser : mettez vos fichiers (RAW ou FITs) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script.

— **OSC_Extract_HaOIII.ssf** : même script que ci-dessus, mais extrait les couches Ha et OIII de l'image couleur. Pour l'utiliser : mettez vos fichiers (RAW ou FITs) dans les dossiers nommés `lights`, `darks`, `flats` et `biases` (dans le dossier de travail par défaut de Siril), puis lancez le script. Vous pouvez aussi utiliser le menu *Traitement de l'image* puis *Composition RVB* et mettre le résultat Ha dans le canal rouge et le résultat OIII dans les couches verte et bleue pour obtenir une image HOO.

> **Astuce**
>
> Pour les possesseurs de filtres **SII** ou dualband **SII-OIII**, les mêmes scripts s'appliquent. En effet, il est impossible pour un capteur couleur de voir la différence entre Ha (656.3 nm) et SII (671.6 nm), tous deux de couleur rouge.

— **RGB_Composition.ssf** : Ce script ajouté dans la version 1.2 aligne des images monochromes avec l'alignement globale, les recadre en utilsiant leur zone commune et prend les trois premières images pour créer une image en couleur. Les images d'entrée doivent être placées seules dans un répertoire et nommées `R.fit` (ou avec l'extension configurée), `G.fit` et `B.fit`. Le résultat est nommé `rgb.fit`. Assurez-vous de supprimer le répertoire `process` entre chaque exécution.

20.1.5 Édition de scripts

Some .ssf script files contain instructions to edit them, for example the Seestar_Preprocessing script instructs the user that if they find too many images are discarded before stacking, they should "increase the value after -filter-round= in the seqapplyreg command, line 47".

The scripts repository directory is not intended as a user-editable directory : at each update it is forcibly reset to match the state of the remote, so files saved in it are not safe. So how to make such changes ?

Find the script in the list in *Preferences->Scripts* and double click it. This will open it in the Script Editor so that you can make the necessary changes and save it in a suitable location (one of the Scripts Storage Directories set at the top of *Preferences->Scripts* is ideal, but it is recommended to give it a different name so you can recognise it in your scripts menu).

> **Astuce**
>
> You can also double click scripts in the list simply in order to examine the code.

Langue de scripts

Au début des scripts, et grâce à la contribution d'un utilisateur, les scripts existaient en deux versions (anglaise, et française). Lors de la sortie de Siril 1.2.0, il a été décidé de ne conserver que les scripts anglais pour des raisons de simplicité de maintenance. Nous encourageons les utilisateurs à distribuer des traductions des scripts officiels à leurs communautés respectives s'ils le jugent nécessaire.

20.1.6 Obtenir plus de scripts

Il y a tout un tas de scripts qui ne sont pas fournis avec l'installation de Siril. Cependant, nous avons mis en place un dépôt gitlab pour eux. Chacun est libre de s'inscrire et de proposer de nouveaux scripts. Nous les accepterons en fonction de leur pertinence : la langue utilisée doit être l'anglais.

Siril intègre git, ce qui signifie qu'il peut télécharger et synchroniser une copie locale du dépôt. Vous pouvez l'activer en cochant la case *Autoriser l'utilisation du dépôt en ligne siril-scripts* dans l'onglet Scripts de la boîte de dialogue Préférences.

En cochant cette case, les scripts seront récupérés dans le dépôt et une liste des scripts disponibles sera affichée, classés en scripts de "Prétraitement" ou de "Traitement". Comme il peut y avoir un nombre important de scripts dans le dépôt, vous devez sélectionner ceux que vous souhaitez avoir dans le menu *Scripts*. Cliquez sur la case à cocher à côté du nom de chaque script que vous souhaitez utiliser, puis cliquez sur le bouton *Appliquer*.

Le contenu d'un script peut être visualisé en double-cliquant sur sa ligne dans la liste. Il est toujours utile de le faire pour vérifier les exigences du script en termes de répertoires préparés en amont. Si vous souhaitez modifier un script, vous pouvez également utiliser cette fonction pour copier le script et le coller dans votre logiciel d'édition de texte préféré. (Vous devrez l'enregistrer dans l'un de vos répertoires de scripts locaux.)

Lorsque le dépôt de scripts est activé, Siril peut synchroniser le dépôt local avec le dépôt distant, soit manuellement, soit automatiquement. Si les mises à jour automatiques sont sélectionnées, la mise à jour aura lieu au démarrage de l'application. La mise à jour manuelle est également disponible en utilisant le bouton *Mise à jour manuelle*. Ce bouton récupère toutes les modifications du dépôt en ligne et affiche une liste des messages de validation décrivant les modifications, que l'utilisateur doit confirmer pour appliquer la mise à jour.

Vous pouvez également vous référer à l'adresse ci-dessous pour parcourir les scripts et les télécharger manuellement si vous ne souhaitez pas utiliser l'intégration git. Dans ce cas, vous devrez placer manuellement les scripts que vous téléchargez dans un chemin de script connu de Siril : https://gitlab.com/free-astro/siril-scripts.

> **Avertissement**
>
> Gardez toutefois à l'esprit que ces scripts ne sont pas nécessairement maintenus par les utilisateurs qui les ont téléchargés et qu'ils peuvent ne pas être à jour. Cela dit, amusez-vous bien.

20.1.7 Rédiger votre propre script

Un fichier script est un simple fichier texte portant l'extension *.ssf.

L'écriture d'un script n'est pas difficile. Il s'agit d'une succession d'appels à des commandes qui sont exécutées séquentiellement. Chaque commande doit être exécutée sans retourner d'erreur, sinon le script s'arrête. Il est donc fortement recommandé d'utiliser la liste des *commandes* pour connaître la syntaxe et le nombre de paramètres nécessaires. Par ailleurs, certaines commandes ne sont pas scriptables et sont signalées par l'icône . Il peut également être utile de tester chaque ligne de script dans la ligne de commande Siril. Vous pouvez lire les scripts fournis ou consulter (voire modifier) les scripts du dépôt à titre d'exemple.

Chaque script doit contenir un en-tête de commentaire contenant des informations sur le script. Un exemple est fourni ci-dessous.

```
##########################################
#
# Script for Siril 1.0
# July 2020
# (C) Cyril Richard
# Mono_Preprocessing_WithoutDark v1.0
#
########### PREPROCESSING SCRIPT ###########
#
# Script for mono camera preprocessing
#
# Needs 3 sets of RAW images in the working
# directory, within 4 directories:
#   biases/
#   flats/
#   lights/
#
##########################################
```

Sous l'en-tête du commentaire, la première commande doit être *requires*. Elle indique la version minimale de Siril requise pour utiliser le script. Par exemple :

```
requires 0.99.4
```

Vous pouvez ensuite lancer le script proprement dit. Les commandes sont placées sur une ligne distincte, et vous pouvez commenter votre script en utilisant des lignes commençant par #.

Chaque nouveau script créé de cette façon devrait être placé dans un *dossier défini par l'utilisateur* pour que Siril puisse les trouver. Si vous pensez que votre script est bénéfique pour la communauté Siril plus large, vous pouvez le soumettre au dépôt de scripts. Les instructions pour ce faire se trouvent dans le README du dépôt.

20.2 Scripts Python

Nouveauté de Siril 1.4, il est possible d'écrire des scripts plus avancés en utilisant Python.

This page is a documentation for script developers. If you want more information about what scripts are available as a user, see the list of scripts.

> **Astuce**
>
> La programmation en Python est encore **EXPÉRIMENTALE** et l'API devrait être améliorée au cours du cycle de développement de la version 1.5.

Dans sa forme actuelle, l'utilisation de scripts Python vous permet de :

- Exécutez n'importe quelle commande Siril tout en bénéficiant de l'avantage d'utiliser un véritable langage de programmation, ce qui permet aux variables transmises aux commandes d'être le résultat de calculs.
- Accéder aux variables et aux structures de données de Siril pour faciliter la rédaction de scripts.
- Accéder aux extensions Python.
  - Using the 1.4 scripting interface it is possible to write a plugin that saves the current image, opens it using astropy and applies processing techniques provided by astropy, then saves the result and reopens it as the Siril main image.
  - L'extension Python **tk** peut être utilisé pour créer des interfaces graphiques pour vos scripts afin de pouvoir recueillir directement les entrées de l'utilisateur.

Avertissement

Dépendances d'exécution

Pour faciliter l'installation et la configuration du module Python Siril et de ses dépendances, un environnement virtuel Python (venv) est configuré.

Linux

This requires that there is a functional Python installation on the system _including the `pip` and `venv` modules. These are available by default as part of the flatpak and appimage distributions, but you will need to install them yourself if building Siril from source. On Debian-based systems you need the `python3-pip` and `python3-venv` packages installed ; the installation requirements on other systems may vary.

Windows

If you have a Python installation in your system, the script will use that as default. If you don't, don't panic. Siril installed through the official installer also comes with a light-weight Python installation and all the required modules. That is for the official releases. If you build the development version from sources, you will need to have Python installed in your system.

macOS

The necessary dependencies are included in the official MacOS package. We do not provide support for homebrew building, but if you do build yourself on Mac then you will need to ensure you have a functional system python installation.

20.2.1 How does this relate to pySiril ?

pySiril is a separate module, published by a separate development team : it is an established product and is already used to implement important helper apps such as Sirilic.

Philosophically, pySiril and the built-in Siril python module are quite different : whereas the built-in module is intended to be called from an already-running Siril to provide scripting functionality and spawns a python3 process to run the script, pySiril is intended for buildin applications that need to call on Siril functionality, and it spawns a Siril process from within its host Python app.

These use cases are quite different, and there is no plan to merge the two modules.

20.2.2 Importer les modules

Siril will take care of setting up a venv (new style python virtual environment) and installing the module in it. In your script you import it using `import sirilpy`.

For convenience, you can abbreviate it as `import sirilpy as s`.

20.2.3 Initialisation de la connexion

Once you have imported the module, you need to establish the interface with Siril. This is done as follows :

```
import sirilpy as s

siril = s.SirilInterface()
try:
    siril.connect()
    print("Connected successfully!")
except SirilConnectionError as e:
    print(f"Connection failed: {e}")
```

(La communication d'informations sur l'état d'avancement est facultative, mais constitue une bonne pratique.)

20.2.4 Exécution des commandes Siril

To execute a Siril command, once the interface is established, you use the `siril.cmd()` function. The command and its arguments are provided as a list, for example :

```
siril.cmd("findstar", "-maxstars=1000")
```

will run the `findstar` command, setting a limit of maximum 1000 stars.

The real power of using python to script commands is that you can use python variables in the command using the formatted string notation :

```
x = 1000
siril.cmd("findstar", f"-maxstars={x}")
```

This is a simple example, but the ability to define parameters to pass to image processing functions is a giant leap for Siril scripting compared with legacy Siril script files.

Astuce

Most Siril commands execute an action (`register`, `stack`, `asinh` etc). These will function exactly the same when called from python.

However some simply print information to the Siril log, such as `stat`. Note that these commands do not behave any differently when called from python using `siril.cmd("stat")` or similar : they will still just print information to the Siril log and do not return any useful data to the python script.

In order to access the values from python you need to use a method that gets information about the image or sequence. So in this example, you would call `img = siril.get_image()` and then the statistics would be available in `img.stats`.

Avertissement

The `cmd()` method does not have a return value but will raise an exception if the command fails. This provides safe default behaviour - the script will stop on failure of a command and print an exception error message to the Siril console. If you want to handle failure more gracefully, for example by showing an error messagebox, you will need to use an exception handler like this :

```
try:
    siril.cmd("requires", "1.5.0")
except:
    siril.error_messagebox("This script is not compatible with this version of Siril")
    quit()
```

For debug purposes you may wish to have commands return a bool. This can be done using a simple wrapper such as the following :

```
def cmd_with_check(*args):
try:
    siril.cmd(*args)
except Exception as e:
    err_msg = ' '.join(list(args))
    print(f'Exception caught: {e}',)
    print(f"Command failed with arguments: {err_msg} but continuing...")
    return False
return True
```

20.2.5 Accès aux variables et aux structures de données de Siril

The siril module provides access to key Siril variables. The available data is as follows :

- Répertoire de travail actuel de Siril
- Répertoire de configuration utilisateur de Siril (destiné à stocker les paramètres spécifiques aux scripts)
- Nom de l'image courant dans Siril
- Image actuelle dans Siril :
  - Données de pixels de l'image
  - Image ICC profile (as bytes : you will need to use a module such as pillow to convert this raw data to something useful)
  - Image metadata : all the metadata that Siril uses internally, including relevant keywords and image statistics. The full FITS header and a string containing keywords not recognized by Siril are also available.
- Séquence Siril actuelle :
  - Sequence frame pixel data
  - Sequence metadata : all the metadata for the currently loaded sequence is available, with the exception of some that relates to details of the sequence format that are abstracted away by the python interface and some relating to photometry series that is not implemented yet. This includes statistics for every channel of every frame, registration data for channels for which it is available of every frame and image quality data for every frame.
  - Sequence frame ICC profiles are not available as these are not generally relevant until after the sequence processing stage. If a convincing use case emerges this could be revisited as an update to the API in the 1.5 development cycle.
  - Sequence frame HISTORY and FITS header strings are not available as they are not considered useful for sequence operations, however the keywords for a sequence frame are available using the `siril.get_seq_frame()` method, which returns a FFit object with the keywords member populated. This provides enough metadata to support any currently identified use case.
- Star modelling data. When stars have been detected in an image the modelling data is available as a list of star parameters for every detected star in the image.

This data is stored in the key data structures `FFit` representing the Siril fits data structure and `Sequence` representing the Siril sequence data structure.

```
# Get the current image
img = siril.get_image()
# Get its dimensions
siril.log(f"Current image dimensions: {img.shape[1]} x {img.shape[0]}, {img.shape[2]}␣
↪channels.")
```

Note that the image shape is stored in a typical python order : shape[0] = height, shape[1] = width, shape[2] = channels. The width, height and channels are also directly accessible using the properties `img.width`, `img.height` and `img.channels` once `img = get_image()` has been called.

When the current image is obtained using `siril.get_image()`, the pixel data can be accessed as a numpy array, allowing directly operating on pixel data.

```
import siril
import numpy as np
# Set up the interface as above
# ...

# Try to get the current image, do something to it and update it in Siril
with siril.image_lock():
    img = siril.get_image()
```

(suite sur la page suivante)

(suite de la page précédente)

```
    img.data[:] *= 2
    siril.set_image_pixeldata(img.data)

except Exception as e:
    raise SirilException(f"Error changing pixel data: {e}")
```

We just accessed the pixel array and multiplied every pixel value by 2! Note that we have to call the `siril.set_image_pixeldata()` function to update the data back to Siril. Using `siril.set_image_pixeldata()` you can even update the image to a different size or shape, or change from 16-bit unsigned data to 32-bit floating point or vice versa.

Astuce

In order for `siril.set_image_pixeldata()` to succeed you must claim a lock on the image. This ensures that nothing else can try to update the image at the same time as you are doing, and also ensures that you can't update the image while something else is already doing so. Note that the example above uses the context handler `siril.image_lock()` **before** getting the image from Siril, not just before updating it. This ensures the image is locked throughout the time it is being processed until it is updated back in Siril and the lock is finally released.

In a similar way you can access image statistics, FITS header information, bit depth and so on. You can also access sequence information, for example whether the *n* th image in a sequence is included, what its stats are and so on.

20.2.6 Installation de modules python

You might have noticed we are using NumPy in the previous examples. This is a dependency of the siril module and will be pulled in during initial setup of the venv. Other python modules can be imported using the standard `import` command, for example `import astropy as ap`. However, for ease of ensuring that dependencies are installed if a user doesn't already have them, the module also provides the `ensure_installed()` method. This is used as follows :

```
s.ensure_installed("astropy")
import astropy
```

(ou, pour un autre exemple,

```
s.ensure_installed("tifffile")
import tifffile
```

First a check is done to try importing the module (in this case, astropy or tifffile). If the module is not available for import, an attempt is made to install it. If the install fails, an exception is raised and the script will halt.

Astuce

If a module requires installation this may take a short time, however the delay only occurs on the first run of a script. On subsequent runs the model is already installed, so the `ensure_installed()` check is almost instantaneous. A message in the log notifies the user that module installation is occurring.

If the check succeeds, the module can be imported as normal using `import`.

Astuce

All other non-core modules except for numpy should be checked with `ensure_installed()` as this automates the installation process for users who don't already have the required modules installed.

Avertissement

All Siril python scripts share the same venv. When importing modules, it is important to avoid overconstraining the version requirements in order to avoid clashes between the modules required by different scripts. Only ">=" version constraints should be used, never "==" or "<=".

```
import sirilpy as s
s.ensure_installed("astropy")
import astropy as ap
```

This will automatically pull in all of astropy's dependencies as well.

20.2.7 Usage de module externe

Astuce

The `siril.FFit` object type is not the same as astropy.io.fits. Whereas astropy.io.fits provides a general file-based interface to all FITS files, siril.FFit provides an interface to the data structure Siril uses to represent FITS images. The two are not directly interchangeable ! Either method can be used to obtain the pixel data as a NumPy array.

Whether you obtain the pixel data directly from the `siril.FFit` object or using astropy.io.fits, siril modules such as astropy, photutils and matplotlib provide building blocks for a huge range of image analysis including source detection, image segmentation, source deblending and visualisation, and the ecosystem of python modules provides limitless capability.

```
import sirilpy as s
s.ensure_installed("astropy")
import astropy
from astropy.io import fits
with fits.open("filename.fit", mode='update') as image:
   if isinstance(image[0].data, np.ndarray):
      image[0].data *= 2
```

This looks much like our last example - all we do is multiply the pixel data by 2 - but now we have used astropy to open it directly from a FITS file.

20.2.8 Référence de l'API

Note that the *API* is very new. While we will try very hard to avoid script-breaking changes to the API released with 1.4.0 throughout the stable 1.4 series, some parts of the API may evolve during the 1.5 development cycle.

20.3 Informations essentielles sur les scripts Python à l'intention des utilisateurs

20.3.1 Obtenir des scripts

Les utilisateurs finaux doivent veiller à se procurer des scripts auprès de sources fiables.

— A few fundamentally important scripts are distributed as part of the core Siril package and can be found installed in the system data directory. These are tested to the same degree as the rest of Siril and the development team will treat bug reports on these scripts in exactly the same way as bug reports on any other part of Siril.

— 1.4.0 introduces the Scripts repository. We try to keep a list of the scripts with a short description here. Anyone may submit scripts to this repository by making an account on gitlab and submitting a merge request. The Siril team will provide a basic level of scrutiny that the scripts are not attempting to do anything malicious, but we do not accept any responsibility for the correct functionality of scripts written by other people and will not provide support for them - the author should be contacted directly.

— Script authors may choose to distribute scripts independently, as has been done in the past. This is fine : scripts can be downloaded and added to a script directory or run via the script editor from anywhere on the filesystem. In this case the Siril team have nothing at all to do with the scripts and cannot even assure you that they will do no harm, so ensure you trust the author.

— Siril does not support automatic verification of signed scripts. This is because script signing can provide a false sense o security ("the script is signed so it can't do anything bad"). This is false : all a script signature says is that the script was signed by a particular person and hasn't been changed since. In other words, it verifies the origin of a script, but says nothing about whether it functions correctly or is safe. **A signed and verified script could still destroy your data if that is what it is written to do !** In the current implementation of python scripting, the majority of scripts are distributed through the centralised script repository in which case the origin is known and scripts have been through a basic level of scrutiny by the development team prior to being merged. Some scripts may be distributed through other channels and downloaded manually, and there is nothing preventing an author providing a sha256sum or a .gpg signature for scripts distributed in this way - you will just need to verify them manually after downloading.

20.3.2 Résolution des problèmes courants

Siril's python script system is quite complex : a lot goes on under the hood, and this is all coordinated by clever use of a Python venv ("virtual environment") which ensures that the python used by Siril is separated from the system Python installation and potentially from other venvs that may be in use by other software. However there are a few things to be aware of :

— venvs are tied to a specific version of python. This means that if your system updates from (e.g.) Python-3.12 to Python-3.13 the venv will no longer point to the right Python program or the right bytecode libraries for compiled modules. **When you upgrade to a new version of Python you must reset the Siril venv**.

— script dependencies are controlled using pip and the pypi archive. This is very good but not 100% foolproof, and in rare cases you may find a dependency issue has arisen that causes a problem of some kind. **If you find an unexplained problem with a script that others aren't experiencing, resetting the venv may solve it.** (And if you find a combination of scripts that can repeatably cause a dependency issue like this, please report it as an issue on the siril-scripts gitlab site as it will help us to improve advice to script authors.)

Resetting the venv is easy : click the *Get Scripts* menu item in the Scripts menu and scroll down to the bottom. Here you will find a "Reset venv" button : clicking it will reset the venv and create a new one. After doing this, the next time you run a script it will need to reinstall its dependencies so the first startup will be a bit slower than ususal (depending on network speed and whether any of the required modules are locally cached).

20.4 Python Script Key Information for Authors

We welcome all script authors who wish to write scripts for Siril. However supporting Siril, while fun, takes a lot of time and we have neither the time nor the desire to support other people's work as well as the core Siril software. There are therefore some guidelines to ensure that the division of responsibility is clear.

— All scripts should feature the author's name or handle, as well as a means of contact (website, YouTube channel, forum etc.) where they can be reached with comments or bug reports. For authors publishing independently this is of course guidance, however for authors wishing to publish their scripts via the siril-scripts repository this is mandatory. Scripts will not be accepted unless these details are included.
— Only open-source scripts will be accepted in the Siril scripts repository. Closed source scripts (including ones that feature a source code wrapper that calls a closed source program) must be independently published.
— Authors should also follow the script programming guidelines on the API page. These guidelines ensure that scripts can be used by end-users with minimal friction.

20.4.1 Normes de codage du dépôt Siril-Scripts

As a script author you can write whatever scripts you like for your own use. However for scripts to be published to the siril-scripts repository we need to require some coding standards in order to ensure that they work with the standard Siril packages on each of the three supported operating systems. The requirements are :

— Scripts should work with versions of Python >= 3.9. This version matches the oldest Python version in our binary packages, and will be adjusted over time to match the packages.
— Scripts should be single-file scripts. The repository was originally designed for legacy .SSF scripts and the assumption of one file per script remains. We understand that it can be good for code readability to develop complex python programs by splitting them into multiple files. This is fine when working locally, however the multiple files would need to be combined into a single file for submission to the scripts repository.
— If you want to implement a GUI in your script you should use the pyQt6 toolkit.
  — pyQt6 uses the well known Qt framework and it has now been verified as installing correctly from pypi on all the supported platforms. Qt6 is much faster than Tkinter for scripts that themselves have to do significant graphics manipulation, such as pan-and-zoom previews, and it has a nice GUI designer application available in Qt Creator.

> **Avertissement**
>
> tkinter is now deprecated : new Siril scripts should not use it and should be written using pyQt6 instead. Although tkinter is a core Python module it is outdated, very slow and has issues with Wayland desktops on Linux. tkinter scripts will continue to work throughout the 1.4 stable series, but there is a considerable amount of added complexity in the sirilpy code to support tkinter. Since pyQt6 provides a much better alternative has now been confirmed to work on all the platforms Siril supports, the tkinter support code (the tksiril and tkfilebrowser submodules) will be removed during the 1.5 development cycle and will be removed before 1.6.0.

> **Avertissement**
>
> Scripts using GTK GUIs will not be accepted into the scripts repository. This GUI framework requires substantial system dependencies. Although Siril itself is coded in GTK, unfortunately the python gi packages required to use it in python scripts are broken on Windows and we are unable to bundle them. If you wish, you can install the dependencies yourself and write scripts using GTK and they may work fine on your own machine, but this is not supported. We will not provide help in getting the prebuilt packages to work with unsupported packages ; if you are determined to try this, you will likely need to build Siril from source.

— Avoid import version specifications that specify a package version using "<", "<=" or "==". These constraints will result in conflicts between scripts, which all share the same venv. Specifying a package version using ">=" is fine. Remember to use `sirilpy.ensure_installed()` before importing non-core python module dependencies, to automate the process of installing them.
— Avoid importing packages that require system packages (i.e. ones that cannot be installed using `python3 -m pip install`)
— Ideally, check that your script imports will install on all three target operating systems. If not then the development team will check them after submission, and you will need to fix any issues that occur.

20.4.2 Modules fonctionnels connus

A list of "known good" packages that we have checked will import without problem, and with all their dependencies, on all the target operating systems, is provided below :

- astropy
- astropy-healpix
- astroquery
- ccdproc
- GaiaXPy
- matplotlib
- numpy
- opencv-python
- pandas
- photutils
- pillow
- pyfftw
- pygaia
- pyQt6
- scipy
- tk
- ttkthemes

Astuce

For machine learning / AI scripts, onnxruntime is very strongly recommended. It is cross-platform, it is hardware-agnostic and it falls back gracefully to CPU support if a GPU or GPU system libraries are unavailable. The `sirilpy.utility` submodule provides an `ONNXHelper` class to ensure the correct hardware- and OS-specific version of onnxruntime is installed.

20.4.3 Module d'apprentissage autonome (machine learning)

A variety of machine learning modules exist which can be used to write "AI" scripts. However the state of support for heterogenous computing (i.e. GPU use) is patchy and often requires system libraries such as CUDA or ROCm to be installed. The following modules are regarded as key for GPU acceleration and either are supported with helper classes or have helper classes in development.

- **onnxruntime** - used in GraXpert and Riccardo's AberrationRemover.py script, this package is available for all the target OSes. There are different backends to support different GPUs, but in some cases these are fragile. A helper class `ONNXHelper()` is available to help with identification and installation of the optimum package as well as ensuring correct runtime fallback to CPU. Note that currently on Windows the ONNXHelper will select the DirectML runtime for all supported GPUs. Even though in theory the CUDA-based onnxruntime-gpu package might be a bit faster, it unfortunately depends heavily on system libraries being installed and configured correctly and the installation is considered too fragile to be reliable in the kind of automated python environment used by Siril. If users wish to use it and are confident they can get it to work on their system they can install it manually using `sirilpy.ensure_installed("onnxruntime-gpu")` but this is not supported and related issues will be closed.
- **torch** - This is a heavyweight among ML frameworks and where GPU support was once very much CUDA-based it is now broadening its coverage : support for NVidia and Apple Silicon GPUs is good, and support for AMD GPUs (using ROCm) and Intel GPUs is in development (ROCm is supposedly stable enough to use on Linux, not yet on Windows). Unfortunately torch has some dependency issues (it is excessively strict about the version of CuDNN it requires) which causes problems using it in the same venv as some other modules (including jax - see below) that pull in the nvidia-cudnn package. A `TorchHelper()` class is available which manages this by first installing torch, then reinstalling it with the --no-deps flag, which prevents it complaining if another package updates the dependencies to a higher version number. This is really ugly and it's unfortunate that it's necessary, but it works. If you envisage using CUDA versions of both torch and jax, it is better to install

torch first and then jax.

— **jax** - Not a dedicated ML framework as such but a means of offloading computation to the GPU. jax provides a mostly numpy-compatible module called jax.numpy which can greatly accelerate many mathematical array processing operations by offloading them to the GPU and also provides functions for gradient computation that numpy does not. Bearing in mind that the slow part of GPU processing is the data transfer between system memory and GPU memory, jax provides a jit compiler to precompile its offloaded workloads but it still requires careful algorithm design to keep data on the GPU for as long as possible, only returning the final result. Jax has good GPU support on Linux but only CPU support on Apple and Windows, although experimental Metal support and Windows GPU support is in progress. This is therefore not recommended for actual scripts at present, but is highlighted as an interesting framework to develop against in the (hopefully) near future when its support broadens. A `JaxHelper()` class is under development that will help with selection of the correct jax package to install and testing to confirm it works properly.

— **Dependencies** Note that both torch and jax use GPU library pypi packages, so installation of them may pull in very large chains of dependencies - the full set of NVIDIA packages (CUDA, CuDNN etc.) is around 1GB. So when using these in scripts be sure to warn users that they may incur a large and potentially slow download on first use !

20.4.4 Modules écartés

Completely avoid using the following modules, as they are known to cause problems on one or more OSes :

— healpy (ne fonctionne pas sous Windows ; on peut utiliser astropy_healpix à la place)

—

20.4.5 Modules présentant des dépendances système

The following modules require installation of system packages which cannot be automated using pip, and therefore should not be directly used in Siril scripts :

— `gi` (pygtk etc - nécessite l'installation d'un paquet système)

—

Avertissement

We cannot emphasise enough that such packages are not supported. If they work for you locally, that's a lucky bonus for you. They generally won't work with any of the prebuilt binary packages (AppImage, flatpak, Windows, MacOS). These kind of packages are specifically not supported and we will not provide help with getting them to work. All tickets relating to this kind of question will be closed as a duplicate of #1527.

Yes, we know it's a shame. In fact we would have liked to use GTK as the toolkit for writing sirilpy scripts, but unfortunately the binary wheel has been broken on Windows for a long time and doesn't show any sign of being fixed. We can't release core functionality that doesn't work on one of the target OSes.

20.4.6 Workaround for Modules with System Dependencies

It may still be possible to write projects that use these kind of packages, but you will need to package them with their dependencies and distribute them independently of Siril or the scripts repository, and the sirilpy module can still be used to achieve integration with Siril. If you want to, you can provide a wrapper script that can be distributed in the scripts repository and just initializes your main program. We will not help with any of that. As stated above, use of these packages in Siril scripts is not supported.

20.4.7 Scripts à code source fermé

The python interface will happily run precompiled .pyc files. Siril is a Free and Open-Source software, and we therefore do not encourage development of closed-source scripts, however we don't forbid it either. Practically we can't, because even stringent interpretations of the GNU Public License allow for writing an open source shim that sits between Siril

and a closed source application. Moreover we are in favour of freedom, and while we choose to release Siril under the GPL to provide freedom to our users, we also respect the freedom of developers to choose how they release their own work.

There is some history here, too : Siril has since 1.2.0 provided a built-in interface to Starnet, however Starnet was originally open source and only closed its doors later on.

However, if an author chooses to release a Siril script as a closed-source .pyc then they need to arrange all matters to do with distributing it : as we are unable to inspect the code ourselves to perform even the most cursory checks, we cannot host it in the Siril-scripts git repository. And of course, responsibility for supporting such products is entirely a matter for the author - the Siril team will offer no advice on such products.

Note that scripts must be compiled separately for each version of python : helloworld.py would be compiled with `python3 -m compileall helloworld.py` which would generate helloworld.cpython-<version>.pyc, where <version> would be 312 for Python 3.12, 313 for Python 3.13 and so on depending on the version of python used to compile the script. This is an inherent inconvenience to closed source scripts.

20.5 Editeur de scripts

Siril contains a script editor, accessible via the Scripts menu. This provides a feature rich code editor tailored for writing both Siril Script Files and Python scripts. The theme (light / dark) of the script editor follows that set in the main *Preferences* dialog. The main editor window is shown below :

For most of the functionality shortcuts exist, which are as standardised as possible.

20.5.1 Menu Fichier

Le menu Fichier offre les fonctionnalités habituelles :

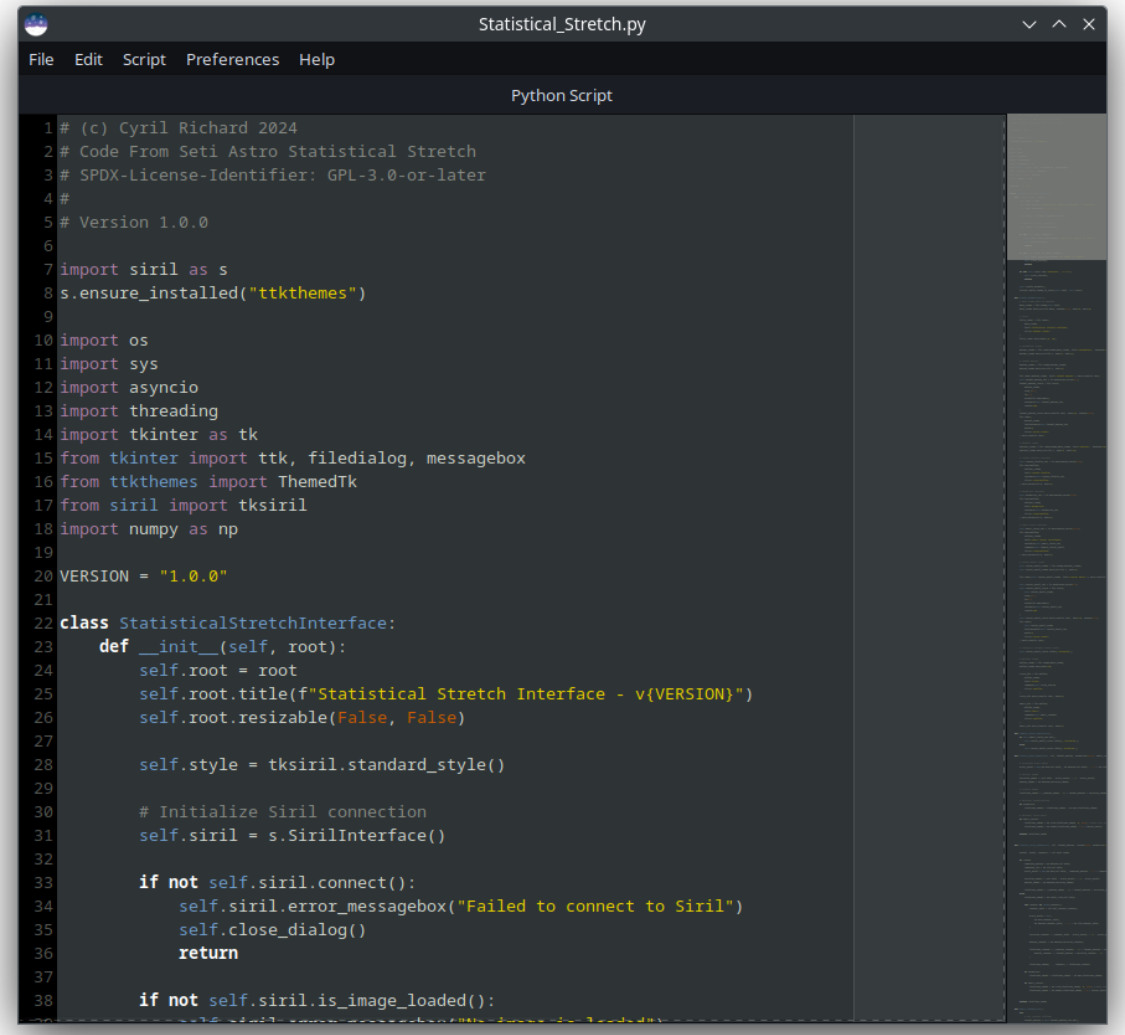
- **New** : begins a new file. If there is unsaved text in the buffer a confirmation dialog will be presented to give the user a chance to save their previous work.
- **Open** : open a file from disk. If If there is unsaved text in the buffer a confirmation dialog will be presented to give the user a chance to save their previous work.
- **Recent files** : provides quick access to open recent files.
- **Save** : saves the current file with its existing filename. If the file is not yet saved then this behaves in the same way as Save as, prompting for a filename.
- **Save as** : saves the current file, prompting for the filename to save it with.
- **Close** : closes the file and the script editor window. If there is unsaved text in the buffer a dialog will be presented to give the user a chance to save their work.

Unsaved files are indicated by an asterisk (*) next to the filename in the titlebar. When changes are saved, the asterisk will disappear.

20.5.2 Menu édition

Le menu Édition propose les fonctionnalités suivantes :

- **Undo** : undoes the most recent change in the script. The script editor has an unlimited undo / redo buffer.
- **Rétablir** : rétabli l'opération la plus récemment annulée.
- **Cut** : cuts the current selection to the clipboard. If no selection is made, the entire current line is cut.
- **Copy** : copies the current selection to the clipboard.
- **Paste** : pastes the current clipboard content to the cursor position.
- **Find** : Presents the *Find overlay* which provides an active search capability. The editor will scroll to the first occurrence of the search term as you type. All occurrences are highlighted in the editor view and in the minimap, if enabled.


Statistical_Stretch.py
File Edit Script Preferences Help
Python Script
(c) Cyril Richard 2024
Code From Seti Astro Statistical Stretch
SPDX-License-Identifier: GPL-3.0-or-later
#
# Version 1.0.0
import siril as s
s.ensure_installed("ttkthemes")
import os
import sys
import asyncio
import threading
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
from ttkthemes import ThemedTk
from siril import tksiril
import numpy as np
VERSION = "1.0.0"
class StatisticalStretchInterface:
def __init__(self, root):
self.root = root
self.root.title(f"Statistical Stretch Interface - v{VERSION}")
self.root.resizable(False, False)
self.style = tksiril.standard_style()
# Initialize Siril connection
self.siril = s.SirilInterface()
if not self.siril.connect():
self.siril.error_messagebox("Failed to connect to Siril")
self.close_dialog()
return
if not self.siril.is_image_loaded():

20.5.3 Menu Script

Le menu du script offre les fonctionnalités suivantes :

- **Run** : runs the current script. Note that this obeys the current script type selection (Siril Script File / Python Script) and this must be correct for the script you are editing. The script type is automatically set based on the file extension when loading a file, but you must set it here if you have not yet saved your script. At startup it defaults to Python Script. Note that owing to a shortcut clash in some desktop environments, as well as the listed `F5`, `Ctrl`-`R` also works to run the current script.
- **Python scripts** : sets the current script mode to Python scripts. This affects the syntax highlighting as well as the manner in which the script is run.
- **Siril Script Files** : sets the current script mode to Siril Script Files. This affects the syntax highlighting as well as the manner in which the script is run.
- **Enable test arguments** : Shows an entry below the main script view. This can be used to enter arguments that will be passed to the script when run from the editor. Python scripts can take arguments to set parameters : this provides an alternative to using a GUI to set parameters. Even if a GUI is provided in a script, the ability to pass arguments means that it can be used as a Siril command via the `pyscript` command, and can therefore itself be used from within scripts.
- **Enable python debug mode** : This toggle changes the behaviour of the sirilpy module to support debugging. When the SirilInterface is created at the start of the script, a modal information box will appear showing the process ID of the python interpreter, which can be used to connect a debugger. Also, all interface timeouts will be set to None, to allow for prolonged inspection of the state of a running script.
  Note that the test arguments have no effect on Siril Script File scripts, as these have no ability to handle arguments.

20.5.4 Préférences

Le menu des préférences propose les options suivantes :

- **Highlight syntax** : when enabled, Siril will use language-aware syntax highlighting to highlight aspects of your script.
- **Enable right margin indicator** : when enabled, Siril will show a right margin indicator at a set column number. This is useful for helping to avoid excessively long lines of code and maintain readability.
- **Right margin position** : this will present a small dialog allowing the user to set the column number of the right margin indicator discussed above. This defaults to 80.
- **Enable bracket matching** : when enabled, if the cursor is on a bracket the editor will highlight the matching bracket. This can be useful to avoid bracket mismatches.
- **Show line numbers** : when enabled, line numbers are shown to the left of the editor view. This can help with debugging and navigating code.
- **Show line markers** : when enabled, line markers are shown to the left of the editor view. No functionality currently uses line marks, but the ability to view them is provided to support future developments.
- **Highlight current line** : when enabled, the current line is highlighted in the editor view.
- **Enable auto-indentation** : when enabled, on pressing Enter the new line will begin at the same level of indentation as the previous line.
- **Indent on tab** : when enabled, pressing tab with a selection made that covers multiple lines will cause all the selected lines to be indented a level, and pressing shift-tab will cause them to be unindented a level.
- **Enable smart backspace** : when enabled, pressing backspace with whitespace to the left of the cursor will delete whitespace back to the previous indentation level.
- **Smart Home / End** : when enabled, Home and End will move the cursor to the first or last non-whitespace character of the line respectively, rather than the absolute first or last character of the line.
- **Show spaces and tabs** : when selected, spaces and tabs will be shown with visible characters (central dots for spaces, right-facing arrows for tab characters).
- **Show newlines** : when selected, newlines will be shown with visible characters (down-and-left bent arrows).
- **Show minimap** : when selected, a minimap will be shown to the side of the editor view, supporting navigation and location of `Find` occurrences.

20.5.5 Aide

The help menu provides API help for both Siril Script Files and Python scripts.

— **Python API Reference** : this opens the Python API page of the online manual in the default browser.
— **Commands Reference** : this opens the Commands Reference page of the online manual in the default browser, which is essentially the API for Siril Script Files.

20.6 Exemple de scripts Python

This section highlights examples of python scripting. These are simple examples intended to provide an introduction to the API, and are explained with commentary.

For a more complete example of a script using a TKinter interface to provide a full-featured interface to an external program, see the CosmicClarity_denoise.py and CosmicClarity_sharpen.py scripts which are available in the Siril scripts repository.

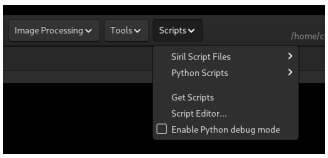


FIG. 3 – Menu script

20.6.1 Hello, Siril !

Il est de tradition de commencer avec un exemple Hello, World !, donc voici le "Hello, Siril !"

```
import sirilpy as s
from sirilpy import SirilConnectionError

siril = s.SirilInterface()

try:
   siril.connect()
   siril.log("Hello, Siril!")

except SirilConnectionError as e:
   print(f"An error occurred: {e}")
```

C'est tout. Tous les scripts Python Siril doivent importer le module siril : celui-ci est fourni automatiquement par Siril et ne nécessite aucune installation.

It allows connection to Siril through the **SirilInterface** class and gives access to Siril methods : in this case the `log()` function does a tiny bit of adjustment (adding a `\n` newline character) and calls the internal function `siril_log_message()` to display the result in Siril's log tab. Note that `log()` can take an optional argument specifying the color : if you wanted "Hello, Siril !" to print in green you would change the line to `siril.log("Hello, Siril!", s.LogColor.GREEN)`.

Note that there is generally no need to call `siril.disconnect()` call, as it is not really necessary : at the end of a script the method will be called automatically in an onexit handler.

Also note that although you can initialize more than one `SirilInterface()` object, a script can only connect via one at a time (though multiple scripts can be connected to Siril simultaneously). There is generally no need to create more than one `SirilInterface()`.

20.6.2 Hello, astropy !

Ce script montre comment enregistrer une copie de travail temporaire, l'ouvrir à l'aide de la fonction astropy.io.fits, la modifier avant d'enregistrer le fichier et le rouvrir dans Siril.

```
import sirilpy as s
import tempfile
import os
import gc
import numpy as np # numpy is a dependency of the siril module and
# will always be available, so no need to use s.ensure_installed() here

s.ensure_installed("astropy")
import astropy
from astropy.io import fits
siril = s.SirilInterface()
try:
   siril.connect():
   print("Connected successfully!")
except SirilConnectionError as e:
   print(f"Connection failed: {e}")
   quit()

def hello_astropy():
   siril.log("Starting the process...")

   temp_filename = None
   try:
      # Create a temporary file
      with tempfile.NamedTemporaryFile(suffix=".fits", delete=False) as temp_file:
            temp_filename = temp_file.name
            siril.log(f"Temporary file created: {temp_filename}")

      # Save current file
      siril.cmd("save", temp_filename)
      siril.log(f"FITS file saved: {temp_filename}")

      # Open and modify FITS file using Astropy
      with fits.open(temp_filename, mode='update') as image:
            siril.log("Opened FITS file with Astropy")

            # Modify the FITS file data
            if isinstance(image[0].data, np.ndarray):
               image[0].data *= 2
               siril.log("Modified FITS file data using Astropy (multiplied pixel values␣
↪by 2)")

            image[0].header['COMMENT'] = "Modified by Astropy"
            siril.log("Added COMMENT to FITS file header")

            image.flush()  # Ensure changes are written
            siril.log(f"Changes saved to FITS file: {temp_filename}")

      # Load back into Siril
```

(suite sur la page suivante)

(suite de la page précédente)

```
      siril.cmd("load", temp_filename)
      siril.log(f"FITS file loaded: {temp_filename}")

      siril.log("Process completed successfully")

   except Exception as e:
      siril.log(f"An error occurred: {str(e)}")
      raise  # Optionally re-raise if needed for further handling

   finally:
      # Clean up: delete the temporary file
      if temp_filename and os.path.exists(temp_filename):
            try:
               os.remove(temp_filename)
               siril.log(f"Temporary file deleted: {temp_filename}")
            except OSError as e:
               siril.log(f"Failed to delete temporary file: {str(e)}")

      # Garbage collection using gc.collect() is not required at the end
      # of a script as it gets run automatically just after the script
      # hands back to the interpreter

# Run the function
if __name__ == "__main__":
   hello_astropy()
```

Notez l'utilisation de la gestion des erreurs dans cet exemple, à l'aide des blocs try : et except : , ainsi que les opérations de nettoyage dans le bloc finally : .

20.6.3 Données de pixels et métadonnées

The sirilpy python module offers direct access to a wide range of data.

```
import sirilpy as s
import numpy as np
siril = s.SirilInterface()
try:
   siril.connect():
   print("Connected successfully!")
except SirilConnectionError as e:
   print(f"Connection failed: {e}")
   quit()

def siril_module_example():
   try:
      # Lock the main image to prevent anything else altering it while we
      # are working on it
      with siril.image_lock():
         # Get the global fits object
         fit = siril.get_image()

         siril.log(f"The value of the first pixel is: {fit.data.flat[0]}")
```

(suite sur la page suivante)

(suite de la page précédente)

```
            fit.data[:] *= 2
            siril.set_image_pixeldata(fit.data)

        siril.log(f"The value of the first pixel is now: {fit.data.flat[0]}")
        siril.log(f"Data type: {fit.data.dtype}")
        siril.log(f"Array shape: {fit.data.shape}")
        history = fit.history
        siril.log(f"History: {history}")

    except AttributeError as e:
        siril.log(f"Error: {e}")
        fit = None  # Ensure the FFit object is released

# Run the function
if __name__ == "__main__":
    siril_module_example()
```

As before, we import the siril module : we now also import numpy, which we will use to operate on the pixel data in the image loaded in Siril. Note that there is no need to use the `import_or_install()` method, as numpy is a dependency of the siril python module and will have been preinstalled in the venv.

Nous allons maintenant présenter différentes façons d'interagir avec les données d'image et les métadonnées :

```
# Get the global fits object
with siril.image_lock():
    fit = siril.get_image(True) # True indicates that we want the pixel data
    # not just the metadata. It is not required as this is the default, but
    # if you only want the metadata you can specify False here to save a bit
    # of memory.

    siril.log(f"The value of the first pixel is: {fit.data.flat[0]}")
    fit.data[:] *= 2
    siril.set_image_pixeldata(fit.data)
    siril.log(f"The value of the first pixel is now: {fit.data.flat[0]}")
```

Nous récupérons l'image actuelle à l'aide de `siril.get_image()`

Astuce

Note that `siril.fits` is not at all the same as `astropy.io.fits`. While astropy.io.fits provides a general-purpose interface to all FITS files, siril.fits provides an interface matching the fits data structure used in Siril.

We then access the pixel data as a NumPy array and carry out operations on this array to modify the image data. In this case we multiply all the pixel values by 2.

After we have finished modifying the image, we call `siril.set_image_pixeldata(fit.data)` to update the pixel data in Siril and trigger the Siril GUI to redraw to show the updated image. Along the way we use `siril.log()` to highlight the change to the first pixel.

Next we show the array datatype and shape :

```
siril.log(f"Data type: {fit.data.dtype}")
siril.log(f"Array shape: {fit.data.shape}")
```

Next we show how we can access the image history. This is provided as a list of individual HISTORY FITS headers, so we join them together and add newline characters before logging.

```
history = fit.history
siril.log(f"History: {history}")
```

Full details of the API can be obtained by importing the Siril module and calling the `help(sirilpy)` command, or more specifically by calling `help()` for the particular part of the API you want to know about, e.g. `help(sirilpy.FFit)`.

Comme les doctsrtings ne sont pas traduisible, pour une traduction de l'aide de l'API vous devez consulter la page de l'API dans ce manuel en ligne.

20.6.4 sous-module siril.plot

This script shows how to use the siril.plot submodule to create a simple plot with three series. The first is a simple x-y plot with lines; the second is an x-y plot with error bars; and the third is another simple x-y plot but shows use of numpy arrays as well as how a SeriesData object can be created and added directly to the PlotData object using `add_series_obj`.

```
import sirilpy as s
import numpy as np
siril = s.SirilInterface()
try:
    siril.connect()
except SirilConnectionError as e:
    print(f"Error: failed to connect to Siril: {e}")
    quit()

# Series data using lists of floats
plot_data = s.PlotData(
    "Example Plot",
    "X Axis",
    "Y Axis",
    "example_plot_lists",
    True,
    datamin = [0.0, 0.0]
    )

plot_data.add_series([1.0, 2.0, 3.0],
                     [4.0, 5.0, 6.0],
                     "Series 1 (Lists)")

plot_data.add_series([4.0, 5.0, 6.0],
                     [7.0, 8.0, 9.0],
                     "Series 2 (Lists with errors)",
                     s.PlotType.MARKS,
                     [0.35, 0.4, 0.45],
                     [0.35, 0.4, 0.45])

x_arr = np.array([1.0, 2.0, 3.0, 4.0], dtype=np.float32)
y_arr = np.array([1.0, 4.0, 9.0, 16.0], dtype=np.float32)
series3 = s.SeriesData(x_arr, y_arr, "Series 3 (np.arrays)")
plot_data.add_series_obj(series3)
```

(suite sur la page suivante)

(suite de la page précédente)

```
# Serialize using lists
siril.xy_plot(plot_data)
```

20.6.5 Modèles de scripts avec interface graphique GUIs

The following templates show how to create scripts with a GUI. The first template is the simplest and only provides a GUI interface :

Code source 1 – gui_template.py

```
# Template for Siril script with GUI only (no CLI)
# SPDX-License-Identifier: # insert license identifier here
# insert author's name and details here

"""
This script template provides an exemplar for use in constructing sirilpy
scripts that provide a Tkinter GUI. Such scripts are only usable via the GUI.
This is a simplified version without command-line argument support.
"""

# Core module imports
import os
import sys
import math
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import numpy as np

import sirilpy as s
from sirilpy import tksiril, SirilError
s.ensure_installed("ttkthemes")
# Add other calls to ensure_installed() here to ensure that any non-core
# modules that are required by the script are installed and available
# to be imported.

# Note: pylint will complain that the following imports are not at the
# top of the module. This deviation from python style is required in order
# to run ensure_installed() to ensure that any non-core modules are
# available before attempting to import them.
from ttkthemes import ThemedTk
# Add any additional imports here

VERSION = "1.0.0"

def template_algorithm(fit,
                       example_float_var,
                       example_bool_var,
                       example_file_path_var):
    """
    Template processing function, insert your algorithm here.
    This is the non-GUI part of the script and is entirely up to you to write.
```

(suite sur la page suivante)

(suite de la page précédente)

```
    Although presented in the template as a single method it may comprise
    classes, methods and anything else.
    """
    print(f"Argument: {example_float_var}")
    print(f"Bool var: {example_bool_var}")
    print(f"File path: {example_file_path_var}")
    return fit

class TemplateScriptInterface:
    """ This class provides the GUI and related callbacks """
    def __init__(self, root):
        self.root = root
        self.root.title(f"Template Script Interface - v{VERSION}")
        self.root.resizable(False, False)
        self.style = tksiril.standard_style()

        # Initialize Siril connection
        self.siril = s.SirilInterface()

        try:
            self.siril.connect()
        except s.SirilConnectionError:
            self.siril.error_messagebox("Failed to connect to Siril")
            return

        # Initial checks: example - check if an image is loaded
        if not self.siril.is_image_loaded():
            self.siril.error_messagebox("No image is loaded")
            return

        # Check if the version of Siril is high enough
        try:
            self.siril.cmd("requires", "1.3.6")
        except s.CommandError:
            return

        # Create the UI and match its theme to Siril
        self.create_widgets()
        tksiril.match_theme_to_siril(self.root, self.siril)

    # This function is used to round values to n decimal places, it is
    # always useful if you have sliders in your UI.
    def _floor_value(self, value, decimals=2):
        """Floor a value to the specified number of decimal places"""
        factor = 10 ** decimals
        return math.floor(value * factor) / factor

    # Here is an example function to round the displayed value of a
    # variable to 2 decimal places. You will need one of these for each
    # slider control you have.
    def _update_example_float_variable_display(self, *args): # pylint: disable=unused-
↪argument
```

(suite sur la page suivante)

(suite de la page précédente)

```
        """Update the displayed template variable value with floor rounding"""
        value = self.example_float_var.get()
        rounded_value = self._floor_value(value)
        self.example_float_var_display_var.set(f"{rounded_value:.2f}")

    def _browse_file(self):
        """
        Use a TK filedialog to browse for a file. Note that this callback is
        specific to the variable being updated; you might need to have several
        similar callbacks if you have more than one file selection widget.
        """
        filename = filedialog.askopenfilename(
            title="Select File",
            initialdir=os.path.expanduser("~")
        )
        if filename:
            self.example_file_path_var.set(filename)

    def create_widgets(self):
        """Create the GUI's widgets, connect signals etc. """
        # Main frame with no padding
        main_frame = ttk.Frame(self.root)
        main_frame.pack(fill=tk.BOTH, expand=True, padx=0, pady=0)

        # Title
        title_label = ttk.Label(
            main_frame,
            text="Template Script",
            style="Header.TLabel"
        )
        title_label.pack(pady=(0, 20))

        # Parameters frame
        params_frame = ttk.LabelFrame(main_frame, text="Parameters", padding=10)
        params_frame.pack(fill=tk.X, padx=5, pady=5)

        ############################################################
        # Example control with an embedded variable in its own frame
        ############################################################

        template_variable_frame = ttk.Frame(params_frame)
        template_variable_frame.pack(fill=tk.X, pady=5)

        ttk.Label(template_variable_frame,
                  text="Template example_float_var:").pack(side=tk.LEFT)

        # Initialize variables with default values
        self.example_float_var = tk.DoubleVar(self.root, value=1.0)
        self.example_float_var_display_var = tk.StringVar(self.root, value="1.00")

        # Add trace to update display when slider changes
        self.example_float_var.trace_add("write",
```

(suite sur la page suivante)

(suite de la page précédente)

```
                                        self._update_example_float_variable_display)

        example_float_var_scale = ttk.Scale(
            template_variable_frame,
            from_=0.0, # set your range minimum here
            to=1.0,    # set your range maximum here
            orient=tk.HORIZONTAL, # oriented horizontally
            variable=self.example_float_var, # the tk variable the widget controls
            length=200
        )
        example_float_var_scale.pack(side=tk.LEFT, padx=10, expand=True)
        ttk.Label(
            template_variable_frame,
            textvariable=self.example_float_var_display_var, # var truncated to 2 d.p.
            width=5,
            style="Value.TLabel"
        ).pack(side=tk.LEFT)
        tksiril.create_tooltip(example_float_var_scale,
                               "Adjusts the template variable.")

        ################################
        # Add frame for other variables
        ################################
        options_frame = ttk.LabelFrame(main_frame, text="Options", padding=10)
        options_frame.pack(fill=tk.X, padx=5, pady=10)

        ##################
        # Example checkbox
        ##################
        self.example_bool_var = tk.BooleanVar(self.root, value=False)
        example_checkbox = ttk.Checkbutton(
            options_frame,
            text="Example checkbox variable",
            variable=self.example_bool_var,
            style="TCheckbutton"
        )
        example_checkbox.pack(anchor=tk.W, pady=2)
        tksiril.create_tooltip(example_checkbox, "Example checkbox.")

        ############################################################
        # Example file selection
        # using an entry and a callback that triggers a filedialog
        # These file selector widgets have their own frame
        ############################################################
        file_frame = ttk.Frame(options_frame)
        file_frame.pack(fill=tk.X, pady=5)

        ttk.Label(file_frame, text="File:").pack(side=tk.LEFT)

        self.example_file_path_var = tk.StringVar(self.root, value="")
        example_file_entry = ttk.Entry(
            file_frame,
```

(suite sur la page suivante)

(suite de la page précédente)

```
            textvariable=self.example_file_path_var,
            width=40
        )
        example_file_entry.pack(side=tk.LEFT, padx=(5, 5), expand=True)

        ttk.Button(
            file_frame,
            text="Browse",
            command=self._browse_file,
            style="TButton"
        ).pack(side=tk.LEFT)

        # The template shows examples of sliders and checkboxes
        # however you can add other sorts of TKinter widgets here

        # Buttons frame
        button_frame = ttk.Frame(main_frame)
        button_frame.pack(pady=20)

        close_btn = ttk.Button(
            button_frame,
            text="Close",
            command=self.close_dialog,
            style="TButton"
        )
        close_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(close_btn,
                               "Close the interface and disconnect from Siril. "
                               "No changes will be made to the current image.")

        apply_btn = ttk.Button(
            button_frame,
            text="Apply",
            command=self.apply_changes,
            style="TButton"
        )
        apply_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(apply_btn,
                               "Apply the script function with the set parameters "
                               "to the current image. Changes can be undone using "
                               "Siril's undo function.")

    def apply_changes(self):
        """ Get the necessary variables from the GUI and call the algorithm """
        try:
            # Get the thread
            with self.siril.image_lock():
                # Get values from the GUI widgets
                example_float_var = self.example_float_var.get()
                example_bool_var = self.example_bool_var.get()
                example_file_path_var = self.example_file_path_var.get()

```

(suite sur la page suivante)

(suite de la page précédente)

```
                # Get current image
                fit = self.siril.get_image()
                fit.ensure_data_type(np.float32)

                # Save original image for undo
                self.siril.undo_save_state(f"Script algorithm: "
                                           "arg={example_float_var:.2f}")

                # Apply script algorithm
                # Your image processing functions go here
                result = template_algorithm(fit, example_float_var, example_bool_var,
                                           example_file_path_var)

                # Clip and update image data
                fit.data[:] = np.clip(result, 0, 1)
                self.siril.set_image_pixeldata(fit.data)

        except SirilError as e:
            messagebox.showerror("Error", str(e))

    def close_dialog(self):
        """ Close dialog """
        self.root.quit()
        self.root.destroy()

def main():
    """ Main entry point """
    try:
        # Create the GUI interface
        root = ThemedTk()
        TemplateScriptInterface(root)
        root.mainloop()
    except SirilError as e:
        print(f"Error initializing script: {str(e)}", file=sys.stderr)
        sys.exit(1)

if __name__ == "__main__":
    main()

language = "fr"
locale_dirs = ["../../translated/"]
gettext_compact = "siril-documentation"
```

Download gui_template.py

The second template is very similar, but provides an alternative interface using a command-line argument vector. This allows the script to be called with arguments using the `pyscript` Siril command : it is useful for scripting algorithms that you might wish to use as part of a bigger scripted workflow.

Code source 2 – gui_and_args_template.py

```
# Template for Siril script with a GUI
# SPDX-License-Identifier: # insert license identifier here
```

(suite sur la page suivante)

(suite de la page précédente)

```
# insert author's name and details here

"""
This script template provides an exemplar for use in constructing sirilpy
scripts that provide a Tkinter GUI but which can also be called using an
argument vector (i.e. so that they can be scripted for use with the Siril
pyscript command). A similar template exists for pure GUI scripts, which
is slightly simpler owing to not having to accommodate command-line
arguments.
"""

# Core module imports
import os
import sys
import math
import argparse
import tkinter as tk
from tkinter import ttk, filedialog, messagebox
import numpy as np

import sirilpy as s
from sirilpy import tksiril, SirilError
s.ensure_installed("ttkthemes")
# Add other calls to ensure_installed() here to ensure that any non-core
# modules that are required by the script are installed and available
# to be imported.

# Note: pylint will complain that the following imports are not at the
# top of the module. This deviation from python style is required in order
# to run ensure_installed() to ensure that any non-core modules are
# available before attempting to import them.
from ttkthemes import ThemedTk
# Add any additional imports here

VERSION = "1.0.0"

def template_algorithm(fit, example_float_var,
                       example_bool_var,
                       example_file_path_var):
    """
   Template processing function, insert your algorithm here.
   This is the non-GUI part of the script and is entirely up to you to write.
   Although presented in the template as a single method it may comprise
   classes, methods and anything else.
   """
    print(f"Argument: {example_float_var}")
    print(f"Bool var: {example_bool_var}")
    print(f"Bool var: {example_file_path_var}")
    return fit

class TemplateScriptInterface:
    """ This class provides the GUI and related callbacks """
```

(suite sur la page suivante)

(suite de la page précédente)

```
    def __init__(self, root=None, cli_args=None):
        # If no CLI args, create a default namespace with defaults
        if cli_args is None:
            parser = argparse.ArgumentParser()
            parser.add_argument("-example_float_var", type=float, default=1.0)
            parser.add_argument("-example_bool_var", type=bool, default=False)
            parser.add_argument("-example_file_path_var", type=str, default="")
            # add other arguments to match the ones in main()
            cli_args = parser.parse_args([])

        self.cli_args = cli_args
        self.root = root

        if root:
            self.root.title(f"Template Script Interface - v{VERSION}")
            self.root.resizable(False, False)
            self.style = tksiril.standard_style()

        # Initialize Siril connection
        self.siril = s.SirilInterface()

        try:
            self.siril.connect()
        except s.SirilConnectionError:
            if root:
                self.siril.error_messagebox("Failed to connect to Siril")
            else:
                print("Failed to connect to Siril")
            return

        # Initial checks: example - check if an image is loaded
        if not self.siril.is_image_loaded():
            if root:
                self.siril.error_messagebox("No image is loaded")
            else:
                print("No image is loaded")
            return

        # Check if the version of Siril is high enough
        # 1.3.6 is the baseline when python scripting became available but in time
        # you may need to check for a higher version number in order to use newer
        # features.
        try:
            self.siril.cmd("requires", "1.3.6")
        except s.CommandError:
            return

        # Create the UI and match its theme to Siril
        if root:
            self.create_widgets()
            tksiril.match_theme_to_siril(self.root, self.siril)

```

(suite sur la page suivante)

(suite de la page précédente)

```
        # Only apply changes if CLI arguments are non-default
        if cli_args and (cli_args.example_float_var != 1.0):
            self.apply_changes(from_cli=True)

    # This function is used to round values to n decimal places, it is
    # always useful if you have sliders in your UI.
    def _floor_value(self, value, decimals=2):
        """Floor a value to the specified number of decimal places"""
        factor = 10 ** decimals
        return math.floor(value * factor) / factor

    # Here is an example function to round the displayed value of a
    # variable to 2 decimal places. You will need one of these for each
    # slider control you have.

    def _update_example_float_variable_display(self, *args): # pylint: disable=unused-
↪argument
        """Update the displayed template variable value with floor rounding"""
        value = self.example_float_var.get()
        rounded_value = self._floor_value(value)
        self.example_float_var_display_var.set(f"{rounded_value:.2f}")

    def _browse_file(self):
        """
        Use a TK filedialog to browse for a file. Note that this callback is
        specific to the variable being updated; you might need to have several
        similar callbacks if you have more than one file selection widget.
        """
        filename = filedialog.askopenfilename(
            title="Select File",
            initialdir=os.path.expanduser("~")
        )
        if filename:
            self.example_file_path_var.set(filename)

    def create_widgets(self):
        """Create the GUI's widgets, connect signals etc. """
        # Main frame with no padding
        main_frame = ttk.Frame(self.root)
        main_frame.pack(fill=tk.BOTH, expand=True, padx=0, pady=0)

        # Title
        title_label = ttk.Label(
            main_frame,
            text="Template Script",
            style="Header.TLabel"
        )
        title_label.pack(pady=(0, 20))

        # Parameters frame
        params_frame = ttk.LabelFrame(main_frame, text="Parameters", padding=10)
        params_frame.pack(fill=tk.X, padx=5, pady=5)
```

(suite sur la page suivante)

(suite de la page précédente)

```
        ################################################################
        # Example control with an embedded variable in its own frame
        ################################################################

        template_variable_frame = ttk.Frame(params_frame)
        template_variable_frame.pack(fill=tk.X, pady=5)

        ttk.Label(template_variable_frame,
            text="Example floting point parameter:").pack(side=tk.LEFT)
        # This is the actual float variable
        self.example_float_var = tk.DoubleVar(self.root,
            value=self.cli_args.example_float_var)
        # This is a string representation of the float var limited to 2 decimal places
        # It is used in the scale label
        self.example_float_var_display_var = tk.StringVar(
            self.root,
            value=f"{self._floor_value(self.cli_args.example_float_var):.2f}")
        # Add trace to update display when slider changes
        self.example_float_var.trace_add("write",
            self._update_example_float_variable_display)

        example_float_var_scale = ttk.Scale(
            template_variable_frame,
            from_=0.0, # set your range minimum here
            to=1.0,    # set your range maximum here
            orient=tk.HORIZONTAL, # oriented horizontally
            variable=self.example_float_var, # the tk variable the widget controls
            length=200
        )
        example_float_var_scale.pack(side=tk.LEFT, padx=10, expand=True)
        ttk.Label(
            template_variable_frame,
            textvariable=self.example_float_var_display_var, # var truncated to 2 d.p.
            width=5,
            style="Value.TLabel"
        ).pack(side=tk.LEFT)
        tksiril.create_tooltip(example_float_var_scale,
            "Adjusts the template variable.")

        ################################
        # Add frame for other variables
        ################################
        options_frame = ttk.LabelFrame(main_frame, text="Options", padding=10)
        options_frame.pack(fill=tk.X, padx=5, pady=10)

        ##################
        # Example checkbox
        ##################
        self.example_bool_var = tk.BooleanVar(self.root,
            value=self.cli_args.example_bool_var
        )
```

(suite sur la page suivante)

(suite de la page précédente)

```
        example_checkbox = ttk.Checkbutton(
            options_frame,
            text="Example checkbox variable",
            variable=self.example_bool_var,
            style="TCheckbutton"
        )
        example_checkbox.pack(anchor=tk.W, pady=2)
        tksiril.create_tooltip(example_checkbox, "Example checkbox.")

        ############################################################
        # Example file selection
        # using an entry and a callback that triggers a filedialog
        ############################################################
        file_frame = ttk.Frame(options_frame)
        file_frame.pack(fill=tk.X, pady=5)

        self.example_file_path_var = tk.StringVar(self.root,
            value=self.cli_args.example_file_path_var)
        example_file_entry = ttk.Entry(
            file_frame,
            textvariable=self.example_file_path_var,
            width=40
        )
        example_file_entry.pack(side=tk.LEFT, padx=(0, 5), expand=True)

        ttk.Button(
            file_frame,
            text="Browse",
            command=self._browse_file,
            style="TButton"
        ).pack(side=tk.LEFT)

        # The template shows examples of sliders and checkboxes
        # however you can add other sorts of TKinter widgets here

        # Buttons frame
        button_frame = ttk.Frame(main_frame)
        button_frame.pack(pady=20)

        close_btn = ttk.Button(
            button_frame,
            text="Close",
            command=self.close_dialog,
            style="TButton"
        )
        close_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(close_btn,
                               "Close the interface and disconnect from Siril. No "
                               "changes will be made to the current image.")

        apply_btn = ttk.Button(
            button_frame,
```

(suite sur la page suivante)

(suite de la page précédente)

```
            text="Apply",
            command=self.apply_changes,
            style="TButton"
        )
        apply_btn.pack(side=tk.LEFT, padx=5)
        tksiril.create_tooltip(apply_btn,
                               "Apply the script function with the set parameters "
                               "to the current image. Changes can be undone using "
                               "Siril's undo function.")

    def apply_changes(self, from_cli=False):
        """
        Get the necessary variables from CLI args or the GUI and call the algorithm
        """
        try:
            # Get the thread
            with self.siril.image_lock():
                # Determine parameters: prefer CLI args if provided,
                # else use GUI values
                if from_cli and self.cli_args:
                    example_float_var = self.cli_args.example_float_var
                    example_bool_var = self.cli_args.example_bool_var
                    example_file_path_var = self.cli_args.example_file_path_var
                else:
                    example_float_var = self.example_float_var.get()
                    example_bool_var = self.example_bool_var.get()
                    example_file_path_var = self.example_file_path_var.get()

                # Get current image
                fit = self.siril.get_image()
                fit.ensure_data_type(np.float32)

                # Save original image for undo
                self.siril.undo_save_state(f"Script algorithm: "
                                           "arg={example_float_var:.2f}")

                # Apply script algorithm
                # Your image processing functions go here
                result = template_algorithm(fit, example_float_var,
                                            example_bool_var,
                                            example_file_path_var)

                # Clip and update image data
                fit.data[:] = np.clip(result, 0, 1)
                self.siril.set_image_pixeldata(fit.data)

            if from_cli:
                print("Script algorithm applied successfully.")

        except SirilError as e:
            if from_cli:
                print(f"Error: {str(e)}")
```

(suite sur la page suivante)

(suite de la page précédente)

```
            else:
                messagebox.showerror("Error", str(e))

    def close_dialog(self):
        """ Close dialog """
        if hasattr(self, 'root'):
            self.root.quit()
            self.root.destroy()

def main():
    """ Main entry point """
    parser = argparse.ArgumentParser(description="Template python script")
    parser.add_argument("-example_float_var", type=float, default=1.0,
                        help="Describe the variable here (0.0 to 1.0)")
    parser.add_argument("-example_bool_var", type=bool, default=False,
                        help="Describe the variable here (bool, default is False)")
    parser.add_argument("-example_file_path_var", type=str, default="",
                        help="Describe the variable here (example file path)")

    args = parser.parse_args()

    try:
        if any([args.example_float_var != 1.0, args.example_bool_var is not False]):
            # CLI mode
            TemplateScriptInterface(cli_args=args)
        else:
            # GUI mode
            root = ThemedTk()
            TemplateScriptInterface(root)
            root.mainloop()
    except SirilError as e:
        print(f"Error initializing script: {str(e)}", file=sys.stderr)
        sys.exit(1)

if __name__ == "__main__":
    main()

language = "fr"
locale_dirs = ["../../translated/"]
gettext_compact = "siril-documentation"
```

Download gui_and_args_template.py

20.7 Current list of Python scripts for Siril 1.4

This page is the first shot at getting a list of Python scripts that is somewhat intelligible and searchable. It was done manually and has only a very succinct description for each script, but will be improved with more words for each with time.

20.7.1 Image processing section

Scripts that process a single image for visual enhancements, in general a stack result.

SyQon/Prism.py	SyQon Prism AI denoise. This is a paying product
SyQon/Starless.py	SyQon Starless AI star removal. This is a paying product
SyQon/Parallax.py	SyQon Parallax AI : aberration correction, star reduction and sharpening. This is a paying product
SyQon/SyQon_Studio.py	New SyQon script that will group all paid models (Parallax, Prism, Starless)
RC-Astro/NoiseXTerminator.py	RC Astro AI denoise. This is a paying product
RC-Astro/StarXTerminator.py	RC Astro Starless AI star removal. This is a paying product
RC-Astro/BlurXTerminator.py	RC Astro Deconv AI : aberration correction, star reduction and sharpening. This is a paying product
processing/Statistical_Stretch.py	Seti Astro Statistical Stretch
processing/AberrationRemover.py	Aberration Remover AI
processing/ContinuumSubtraction.py	Narrowband Continuum Subtraction script
processing/DeepSNR.py	DeepSNR AI denoise (wrapper around the third-party DeepSNR tool)
processing/StarNet.py	StarNet 2.5 star removal (wrapper around the third-party StarNet tool), with optional star mask generation
processing/GPS_Process.py	Image processing based on CosmicClarity and GraXpert (background extraction, denoise and/or sharpening)
processing/Narrowband_Palette_Picker.p	Helps preview different palettes for narrowband image composition
processing/GraXpert-AI.py	Integration of the GraXpert gradient removal tool (AI)
processing/CosmicClarity_Denoise.py	Cosmic Clarity Denoise
processing/DBXtract.py	Extract the Sulfur II, Hydrogen Alpha and Oxygen III signal from dual band filters to compose SHO images in color cameras
processing/Hubble_Palette_from_Band_OSC.py	Create different "Hubble-like" palettes from OSC (One-Shot Color) images acquired with a dual-band Ha/OIII filter
processing/PalettePicker.py	Script that help user to create its own palette for narrowband filters
processing/HDR_multiscale.py	Wavelet-Based Dynamic Range Compression
processing/HDR_Blender.py	Inverse Variance Maximum Likelihood Estimate (MLE) HDR Blender. Blends multiple images with different exposures
processing/NB_2_RGB.py	Monochrome image combination into a color image
processing/SCUNet_Denoise.py	SCUNet image denoiser (AI)
processing/CosmicClarity_Superres.py	Cosmic Clarity Superres
processing/CosmicClarity_Native.py	Cosmic Clarity AI-powered sharpening, denoising, super resolution and star removal
processing/AutoBGE.py	Auto Background Extraction script
processing/AutoGradientRemoval.py	Auto Gradient Removal tool. This script will be implemented in the Siril code in the next version
processing/CosmicClarity_Satellite.py	Cosmic Clarity Satellite Removal
processing/CosmicClarity_Darkstar.py	Cosmic Clarity Darkstar
processing/ER-Bill_Star_Reduction.py	Script for reducing stars using pixel math

suite sur la page suivante

Tableau 1 – suite de la page précédente

processing/DSA-Star_Reduction.py	Script for reducing stars using pixel math
processing/CosmicClarity_Sharpen.py	Cosmic Clarity sharpening process
VeraLux/VeraLux_Alchemy.py	Linear-Phase Narrowband Normalization & Mixing
VeraLux/VeraLux_Nox.py	Physically-Faithful Photometric Gradient Reduction
VeraLux/VeraLux_StarComposer.py	High-Fidelity Star Reconstruction Engine
VeraLux/VeraLux_Revela.py	Photometric Local Contrast & Texture Engine
VeraLux/VeraLux_HyperMetric_Stret	Photometric Hyperbolic Stretch Engine
VeraLux/VeraLux_Starting_Point.py	Interactive Workflow Guide & Manual
VeraLux/VeraLux_Curves.py	Spline-Based Photometric Sculpting Engine
VeraLux/VeraLux_Silentium.py	Linear-Phase Noise Suppression Engine
VeraLux/VeraLux_Vectra.py	Vector Color Grading & Chromatic Surgery Engine

20.7.2 Sequence pre-processing section

Scripts that process a set of images, an alternative to usual preprocessing scripts.

preprocessing/AMSP.py	Siril Wizard – Automatic Multi-Session Processing. Drag and drop all files and it does the stacks
preprocessing/StorageFriendlyStacking.py	Storage Friendly Stacking Script
preprocessing/Naztronomy-Smart_Telescope_PP.py	Naztronomy - Smart Telescope Preprocessing script
preprocessing/GPS_Preprocess.py	Preprocessing with multi-session, mosaics and many optional steps support
preprocessing/ER-CometStartrail.py	Stack a startrail from a sequence
preprocessing/osc-multi-night-stacking-v1.2.py	Multi-night stacking
preprocessing/Naztronomy-OSC_PP.py	Naztronomy - OSC Image Preprocessing script with mosaics support

20.7.3 Utility section

Scripts that provide some tool for data analysis or other non-processing related tasks.

core/Siril_Catalog_Installer.py	Siril Catalog Installer
core/GPU_Manager.py	A GUI tool for managing ONNX, PyTorch, and JAX installations
utility/Selected_Star_Spectrum.py	Plots the Gaia DR3 continuous spectrum for any selected Gaia DR3 source
utility/RegistrationInspector.py	Displays the frames of selected images to check framing
utility/SuperStack.py	Performs superstacking (moving average)
utility/plot3D.py	Plots the current image or selection in 3D using matplotlib
utility/Sequence_Statistics_Analyzer	Analyze frames and plot key statistics

suite sur la page suivante

Tableau 2 – suite de la page précédente

utility/Workflow_Companion.py	Deep Space Astro drag-and-drop workflow companion : queue, reorder and launch scripts and Siril functions
utility/Workflow_Summarizer.py	Uses Google Gemini to summarize the Siril log into readable workflow documentation
utility/Blink_Browse_Filter_Sort.py	Image browser / filter / sorter with adaptive caching
utility/Svenesis-GradientAnalyzer.py	Gradient Analyzer
utility/Svenesis-MultipleHistogramViewer.py	Multiple Histogram Viewer
utility/Svenesis-CosmicDepth3D.py	CosmicDepth 3D
utility/Patch_Inpainting_Tool.py	Corrects defects and creates a star mask
utility/Svenesis-AnnotateImage.py	Full size image annotation
utility/AF_Multi_Crop.py	Multi-crop script
utility/Svenesis-BlinkComparator.py	Blink Comparator
utility/Satellite_Trail_Removal.py	Remove satellite trails
utility/Sequence_Deleter.py	GUI to delete sequences
utility/ImageWindow.py	Pseudo-MDI GUI for storing and swapping images
utility/Dwarfium_Archive_Selector.p	Prepare sessions from Dwarf telescopes
utility/AstroT3kFetch.py	Fetch and classify astrophotography frames
utility/Diffraction_Spike_Overlay.py	Add diffraction spikes
utility/Flat_On_Flat_Analyzer.py	Analyze flat field effectiveness
utility/Distortion3D.py	Plot 3D distortion map
utility/Autocrop.py	Autocrop stacked images
utility/Signature_Tool.py	Insert a signature/logo
utility/Galaxy_Annotations.py	Create galaxy annotations from Simbad queries
utility/Asteroid_Comet_Finder.py	Search for asteroids and comets
utility/AutoStretch_Preview.py	Interactive AutoStretch preview
utility/HertzsprungRussell.py	Create a Hertzsprung-Russell diagram

Avertissement

Tous les scripts ne sont pas rédigés, ou ne relèvent de la responsabilité, de l'équipe de développement de Siril !

Avec l'introduction des scripts Python, les possibilités offertes aux auteurs de scripts Siril se sont considérablement élargies. Un grand pouvoir implique de grandes responsabilités !

You, as the user, need to ensure you trust the authors of scripts you use. 1.4.0 introduces the script repository. Some of the scripts there have been written by the Siril development team but others have been written by third party contributors. Please read the Key Info section regarding guidance for script authors, end users and the rules for the script repository.

20.8 Obtenir des scripts

A small number of basic script files come packaged with Siril, however you can add more from the script repository. To do so, go to the *Scripts menu -> Get Scripts*. This will open the *Scripts* tab of the *Preferences* dialog. Here, below the list of local script directories, you will find a list of scripts in the repository.

Les scripts peuvent être triés par catégorie, par nom, par type ou selon qu'ils sont sélectionnés ou non.

As there may eventually be a large number of scripts, not all of which will be relevant to your workflow, you can configure the scripts you want to appear in the *Scripts* menu. Simply select the checkbox next to them and click Apply to update the menu.

Astuce

You don't need to select scripts to appear in the menu for them to be usable with the `pyscript` command : this will search all the local script directories as well as the whole of the local scripts repository.

Avertissement

Script authors may submit updates. Occasionally, inevitably, there may be a bad update that breaks the script for you. Bugs need to be reported to the script author : for scripts written by the Siril team you can report them at the scripts GitLab repository and for those written by third party contributors you should report them at the contact details in the script.

Le script peut être ouvert dans l'éditeur de script en double-cliquant dessus dans la liste. Ceci est important pour plusieurs raisons :

— Cela vous permet de trouver les coordonnées de l'auteur du script afin de lui signaler un bug.
— Lorsque vous double-cliquez sur le script, une option vous permet de revenir aux versions précédentes.
— This means that if an update broke the script, you can open the previous version and save a local copy in one of your local script folders for use until the script author fixes the bad update. (The version saved in a local folder will take priority over a repository version with the same name, but it will appear in a different subfolder in the menu with the name of your local scripts folder.)

CHAPITRE 21

Mode console

Siril peut fonctionner à la fois avec son interface graphique (GUI) et avec une interface en ligne de commande (CLI) qui ne nécessite même pas d'affichage. Il peut traiter des images pour d'autres programmes, sur des ordinateurs distants ou embarqués, en utilisant des scripts ou des opérations générées en temps réel appelées commandes. Les capacités de Siril en mode console correspondent en fait à celles des *commandes disponibles*. Il y en a plus d'une centaine, permettant de réaliser automatiquement des prétraitements, des traitements et des analyses photométriques.

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.

Avec la version sans interface graphique, les commandes peuvent être exécutées soit en passant un script à exécuter, soit en définissant l'entrée standard comme un script et en écrivant des commandes dedans, avec l'option de ligne de commande `-s -`, ou en utilisant des tubes nommés.

Voici un exemple de code bash appelant le mode console, qui construit le master-bias et renvoie son niveau de bruit sur la console, en rouge :

```
#!/bin/bash
# bash commands to prepare files

initdir=$(pwd)

######## Set your own variables #############
SCRIPTS_DIRECTORY=$initdir
SIRIL_PATH=siril-cli
#############################################

# Removing process folder if exists #
rm -rf $SCRIPTS_DIRECTORY/process

echo "Running siril bash script in $initdir"
output=$($SIRIL_PATH -d $SCRIPTS_DIRECTORY -s - <<ENDSIRIL
```

(suite sur la page suivante)

(suite de la page précédente)

```
##########################################
#
# Example of script that create a master-bias
# and print in red the noise of the image
#
##########################################

requires 1.0.0

# Convert Bias Frames to .fit files
cd biases
convert bias -out=../process
cd ../process

# Stack Bias Frames to bias_stacked.fit
stack bias rej 3 3 -nonorm -out=master-bias
cd ../..

close
ENDSIRIL
)

log_line=$(echo "$output" | grep -o "log: Background noise value.*")
echo -e "\e[31m$log_line\e[0m"

echo done Siril part
```

21.1 Flux de commande (Pipe)

Ce mode a été introduit avec Siril 0.9.10. Les commandes peuvent être envoyées via un tube nommé et les journaux et l'état peuvent être obtenus via un autre. Le mode est activé en utilisant l'argument en ligne de commande `-p`.

Le protocole est assez simple : Siril reçoit des commandes et renvoie des mises à jour. Seules les commandes marquées comme scriptables peuvent être utilisées avec ce système. Chaque fois que le tube de commande d'entrées est fermé ou que la commande d'annulation est donnée, la commande en cours d'exécution est arrêtée comme si le bouton d'arrêt avait été cliqué dans l'interface graphique. Les tubes sont nommés `siril_command.in` et `siril_command.out` et sont disponibles dans `/tmp` sur les systèmes Unix.Depuis la version 1.2.0, les chemins des tubes peuvent être configurés avec les options `-r` et `-w`, ce qui permet à des programmes externes de les créer avant de démarrer Siril, généralement avec la commande `mkfifo`. Également nouveau dans la version 1.2.0, une commande `ping` renvoie simplement un état, indiquant si Siril est prêt à traiter une commande ou déjà occupé.

Les sorties de Siril sur le tube sont un flux d'une ligne de texte et sont formatées comme suit :

- `ready` est affiché au démarrage, indiquant que Siril est prêt à traiter les commandes
- `log:` suivi d'un message de journalisation
- `status: verb [subject]`, où le verbe peut être `starting`, `success`, `error` ou `exit` (le message de sortie n'est pas encore implémenté). Le sujet est le nom de la commande en cours, sauf pour sortie qui indique que Siril a subi une erreur fatale et doit quitter.
- `progress: valeur%` est l'équivalent de la barre de progression, il envoie périodiquement des pourcentages, et parfois 0% à la fin d'un traitement en tant qu'information de veille.

CHAPITRE 22

Interprétation d'entête

Le parsing est la capacité d'analyser, c'est-à-dire d'écrire des chaînes de caractères à partir des données contenues dans l'en-tête *FITS*. L'interprétation d'entêtes, introduit dans Siril 1.2.0, vise à donner plus de flexibilité au scripting en utilisant les données de l'entête pour écrire/lire des noms de fichiers ou des chemins. Pour l'instant, ceci est utilisé avec les commandes suivantes :

- *calibrate*
- *stack* et *stackall*
- toutes les commandes *saveXXX*

et bien sûr, leurs équivalents GUI.

22.1 Exemple de syntaxe

Nous allons prendre un exemple simple pour commencer. Supposons que vous ayez un fichier nommé `light_00001.fit` et que vous souhaitiez trouver un masterdark dans votre bibliothèque de masters qui corresponde aux caractéristiques de cette image. Comme vous avez choisi une convention pour nommer vos masterdark, vous savez que le bon dark doit être nommé comme suit :

```
DARK_"exposure"s_G"gain"_O"offset"_T"temperature"C_bin"binning".fit
```

avec les termes entre parenthèses remplacés par les valeurs lues dans votre en-tête d'image brute. Pour une exposition de 120s, une température de -10C, un gain/offset de 120/30 et un binning 1, le masterdark serait nommé :

```
DARK_120s_G120_O30_T-10C_bin1.fit
```

C'est exactement ce que cette fonctionnalité permet de faire. Si vous spécifiez le nom du dark avec les conventions expliquées juste après, vous pouvez dire à Siril d'ouvrir l'image, de lire son en-tête et d'utiliser ses valeurs pour écrire cette chaîne (et ensuite l'utiliser pour calibrer votre images).

You can read the info contained in the header either through *dumpheader* command or by using the Tools menu, *Tools → FITs Header...* ` You would normally get a print like the one below (some keys removed for brevity) :

```
SIMPLE  =                    T / C# FITS
BITPIX  =                   16 /
```

(suite sur la page suivante)

(suite de la page précédente)

```
NAXIS   =                    2 / Dimensionality
NAXIS1  =                 4144 /
NAXIS2  =                 2822 /
BZERO   =                32768 /
EXTEND  =                    T / Extensions are permitted
IMAGETYP= 'LIGHT'              / Type of exposure
EXPOSURE=                120.0 / [s] Exposure duration
DATE-OBS= '2022-01-24T01:03:34.729' / Time of observation (UTC)
XBINNING=                    1 / X axis binning factor
YBINNING=                    1 / Y axis binning factor
GAIN    =                  120 / Sensor gain
OFFSET  =                   30 / Sensor gain offset
INSTRUME= 'ZWO ASI294MC Pro'   / Imaging instrument name
SET-TEMP=                -10.0 / [degC] CCD temperature setpoint
CCD-TEMP=                -10.0 / [degC] CCD temperature
BAYERPAT= 'RGGB'               / Sensor Bayer pattern
TELESCOP= '61EDPH'             / Name of telescope
FILTER  = 'DualBand'           / Active filter name
OBJECT  = 'Rosette Nebula     '/ Name of the object of interest
OBJCTRA = '06 30 36'           / [H M S] RA of imaged object
OBJCTDEC= '+04 58 51'          / [D M S] Declination of imaged object
RA      =     97.6960081674312 / [deg] RA of telescope
DEC     =     4.99212765957446 / [deg] Declination of telescope
END
```

Le format utilisé pour spécifier le nom des darks serait alors :

```
DARK_$EXPTIME:%d$s_G$GAIN:%d$_O$OFFSET:%d$_T$SET-TEMP:%d$C_bin$XBINNING:%d$.fit
```

Tous les termes à analyser sont formés comme suit : $**KEY** :**fmt**$

- **KEY** est n'importe quelle clé (valide) de l'en-tête FITS de l'image brute
- **fmt** est un spécificateur de format.

Par exemple, `$EXPTIME:%d$` sera interprété comme `120` si la brute a été exposée pendant 120s. Mais il sera interprété comme `120.0` si vous spécifiez `$EXPTIME:%0.1f$`, grâce au formateur `%x.yf`.

L'expression complète ci-dessus serait donc évaluée à :

```
DARK_**120**s_G**120**_O**30**_T**-10**C_bin**1**.fit
```

Dans ce premier exemple, nous n'avons utilisé que la conversion en nombres entiers avec `%d`. Mais il existe d'autres formatages conventionnels que vous pouvez utiliser :

- `%x.yf` pour les nombres flottants
- `%s` pour les chaînes de caractères

Note

Pour les chaînes de caractères, les espaces devant et derrière sont toujours supprimés, tandis que les espaces à l'intérieur des chaînes sont remplacés par des signes _. Exemple : `$OBJECT:%s$` sera converti en `Rosette_Nebula`.

Vous pouvez aussi utilisé des formatages moins conventionnels :

- Pour analyser une date à partir d'un mot clé d'en-tête date-heure, vous pouvez utiliser le format spécial non standard **dm12**, qui signifie date moins 12h. Dans l'en-tête ci-dessus, la clé DATE-OBS a pour valeur

2022-01-24T01:03:34.729. La valeur de $DATE-OBS:dm12$ serait convertie en 2022-01-23, ce qui correspond à la date du début de la nuit. Vous pouvez également utiliser le formateur spécial **dm0** qui analysera simplement la date, sans soustraire 12h.

— Pour analyser une date-time à partir d'une clé d'en-tête date-time, vous pouvez utiliser le format spécial non standard **dt**, qui signifie simplement date-time. Dans l'en-tête ci-dessus, le mot clé DATE-OBS a pour valeur 2022-01-24T01:03:34.729. La conversion de $DATE-OBS:dt$ se ferait alors en 2022-01-24_01-03-34.

— Pour analyser les informations RA et DEC des mots clés d'en-tête OBJCTRA et OBJCTDEC, vous pouvez utiliser les formateurs spéciaux non standard **ra** et **dec**. Dans l'en-tête ci-dessus, les mots clés OBJCTRA et OBJCTDEC ont une valeur de 06 30 36 et +04 58 51 respectivement. $OBJCTRA:ra$_$OBJCTDEC:dec$ serait converti en 06h30m36s_+04d58m51s.

— Pour analyser les informations RA et DEC à partir des mots clés d'en-tête RA et DEC, lorsqu'elles sont au format décimal, vous pouvez utiliser les formats spéciaux non standard **ran** et **decn**. Dans l'en-tête ci-dessus, les mots clés RA et DEC ont une valeur de 97.6960081674312 et 4.99212765957446 respectivement. La conversion de $RA:ran$_$DEC:decn$ serait 06h30m47s_+04d59m32s.

Un bon exemple de noms de fichiers pour un résultat d'empilement est l'expression suivante :

```
$OBJECT:%s$_$FILTER:%s$_$STACKCNT:%d$x$EXPTIME:%d$sec_G$GAIN:%d$_O$OFFSET:%d$_T$CCD-TEMP:
↪%d$°C_$DATE-OBS:dm12$
```

en donnant quelque chose comme :

```
NGC_7023_L_57x120sec_G100_O50_T-9°C_2023-10-07
```

Pour tester la syntaxe, vous pouvez charger une image et utiliser la commande *parse*, comme rappelé ci-dessous.

Ligne de commande Siril

```
parse str [-r]
```

Analyse la chaîne de caractères **str** en utilisant les informations contenues dans l'en-tête de l'image actuellement chargée. Le but principal de cette commande est de débuguer l'interprétation d'entête des clés d'en-tête qui peuvent être utilisées dans d'autres commandes.

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.

Si la chaîne **str** commence par le préfixe *$def*, elle sera reconnue comme un mot-clé réservé et recherchée dans les chaînes stockées dans gui_prepro.dark_lib, gui_prepro.flat_lib, gui_prepro.bias_lib ou gui_prepro.stack_default pour *$defdark*, *$defflat*, *$defbias* ou *$defstack* respectivement.

Le mot-clé *$seqname$* peut également être utilisé quand une séquence est chargée

22.2 Trouver un fichier avec l'interprétation d'entête

Dans l'exemple ci-dessus, nous avons vu que nous pouvions trouver le nom d'un dark maître à partir des informations contenues dans l'en-tête de la brute à calibrer. C'est ce que l'on appelle dans la commande `parse`, le `mode lecture`.

This behavior is mainly used in conjuction with the *calibrate* command/tab and to find master distortion files. 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.

Par exemple, pour les flats, vous pouvez vouloir spécifier le chemin d'accès à une bibliothèque, qui pourrait contenir des informations sur les filtres ou les télescopes, car vous pouvez avoir plusieurs configurations. Un chemin comme :

```
~/astro/masters/flats/$INSTRUME:%s$_$TELESCOP:%s$/$FILTER:%s$/FLAT_bin$XBINNING:%d$.fit
```

serait alors évalué comme :

```
~/astro/masters/flats/ZWO_ASI294MC_Pro_61EDPH/DualBand/FLAT_bin1.fit
```

et est une valeur valide pour l'entrée `Flat`.

Bien sûr, si vous deviez écrire cette commande dans vos scripts ou dans le champ "Flat" de l'interface graphique à chaque fois que vous calibrez, cela pourrait devenir un peu fastidieux. C'est là que les mots-clés réservés viennent à la rescousse. Il y a 3 mots-clés réservés pour les maîtres :

- `$defdark`
- `$defflat`
- `$defbias`
- `$defdisto`

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`, `gui_prepro.bias_lib`. and `gui_prepro.disto_lib`.

Lorsque leurs valeurs sont définies et que vous avez choisi de les utiliser comme valeurs par défaut, elles seront affichées dans les champs de l'onglet *Calibration*. Vous pouvez également commencer à écrire vos scripts en utilisant ces mots-clés. L'étape de calibration d'un nouveau script générique pour une caméra couleur pourrait ressembler à ceci :

```
calibrate light -dark=$defdark -cc=dark -flat=$defflat -cfa -equalizecfa -debayer
```

Cela permet de prélever vos masters directement dans vos bibliothèques et de s'assurer que vous ne les mélangez jamais pendant l'étape d'étalonnage.

22.3 Écriture d'un fichier avec interprétation de l'entête

Maintenant, bien qu'il soit pratique de pouvoir trouver les fichiers, il serait tout aussi utile d'utiliser cette syntaxe pour stocker vos fichiers pendant l'empilement. C'est exactement ce à quoi sert le `mode écriture`. La syntaxe peut être utilisée dans le champ `-out=` des commandes *stack* et *stackall*, ou dans le champ correspondant dans l'interface graphique.

Imaginons que vous vouliez écrire un script générique qui prépare vos master darks à chaque fois que vous renouvelez votre bibliothèque. Dans la ligne `stack` du script, vous pourriez écrire :

```
stack dark rej 3 3 -nonorm -out=$defdark
```

Cette ligne garantit que le masterdark résultant sera stocké au bon endroit avec le nom de fichier correct qui peut ensuite être récupéré pour calibrer vos brutes.

Afin d'introduire encore plus de flexibilité avec les commandes `stack`, il y a deux mots-clés réservés supplémentaires :

- `$defstack`
- `$sequencename$`

Comme pour les maîtres par défaut, `$defstack` est configuré dans la même section des Préférences, ou avec une commande *set* sur `gui_prepro.stack_default`. Par exemple, supposons que vous ayez défini `$defstack` comme :

```
Result_$OBJECT:%s$_$DATE-OBS:dm12$_$LIVETIME:%d$s
```

La ligne de script :

```
stack r_pp_light rej 3 3 -norm=addscale -output_norm -out=$defstack
```

sauvegarde le résultat de l'empilement comme :

```
Result_Rosette_Nebula_2022-01-24_12000s.fit
```

Depuis Siril 1.2.0, le nom par défaut pour l'empilement est défini comme `$sequencename$stacked` (le signe _ est ajouté s'il n'est pas présent). Ce comportement est similaire à celui des versions précédentes, sauf qu'il est maintenant explicite que l'interprétation de l'entête est utilisée.

22.4 Utilisation des caractères de remplacement

Il se peut que vous vouliez utiliser une valeur clé dans le nom de vos masters qui ne correspond pas à la valeur clé des images à calibrer. Avec un exemple, ce sera peut-être un peu plus clair. Disons que vous voulez, dans le nom de vos masters, enregistrer leur temps d'exposition. Quelque chose comme :

```
FLAT_1.32s_Halpha_G120_O30_bin1.fit
```

Si vous mettez un champ $EXPTIME :%0.2f$ dans `$defflat`, cela se terminera par une erreur à l'étape de calibration. Tout simplement parce que la clé EXPTIME sera lue à partir de l'image brute à calibrer, et non d'un flat.

Pour faire face à cette situation, vous pouvez utiliser des caractères génériques dans les expressions à analyser :

```
FLAT_$*EXPTIME:%0.2f$_$FILTER:%s$_G$GAIN:%d$_O$OFFSET:%d$_bin$XBINNING:%d$
```

Notez le symbole * placé juste avant `EXPTIME`.

La signification de ce symbole est la suivante :

- En `mode écriture`, c'est-à-dire lors de l'empilement de votre flat maître, le champ `EXPOSITION` sera utilisé pour former le nom du fichier à sauvegarder. Dans l'exemple ci-dessus, vous sauvegarderez effectivement le fichier `FLAT_1.32s_Halpha_G120_O30_bin1.fit`.
- En `mode lecture`, donc lors de l'étalonnage de vos brutes, le champ `EXPOSURE` sera remplacé par *. Lors de la recherche d'un tel fichier, Siril récupérera tous les fichiers qui correspondent au modèle `FLAT_*_Halpha_G120_O30_bin1.fit`. Avec un peu de chance, votre convention de nommage est suffisamment robuste pour qu'il ne trouve qu'un seul fichier correspondant et l'utilise pour le calibrage.
  Cela peut également s'avérer utile pour définir des fichiers de distorsion maîtres. Vous pourriez, par exemple, utiliser :

```
~/astro/masters/distos/DISTO_$*DATE-LOC:dm12$_$INSTRUMEN:%s$_$TELESCOP:%s$_bin$XBINNING:
↪%d$.wcs
```

Notez le symbole * placé juste avant `DATE-LOC`.

When a wildcard is used in front of a `DATE` header key, the returned file in readmode is the file with the closest date smaller or equal to the file which header is being parsed.

Avertissement

Dans le cas où Siril trouve plus d'un fichier en "mode lecture", il émettra un avertissement dans la console et sélectionnera le fichier le plus récent. Comme cela peut ne pas produire le résultat souhaité, vous devriez reconsidérer votre convention de nommage si cela se produit.

22.5 Gestion des doublons

Dans certains cas, il existe plusieurs mots-clés pour la même valeur. En effet, les développeurs de logiciels sont libres d'utiliser les mêmes mots-clés ou d'en créer de nouveaux. Siril tente donc de reconnaître et de gérer les doublons. Voici un tableau récapitulatif des doublons connus. Si un fichier contient des mots-clés dits « alternatifs », alors Siril stockera la valeur dans la version « primaire ».

Mot-clé principal	Alternatif
MIPS-HI	CWHITE
MIPS-LO	CBLACK
PROGRAM	SWCREATE
IMAGETYP	FRAMETYP
EXPTIME	EXPOSURE
FILTER	FILT-1
FOCALLEN	FOCAL
CENTALT	OBJCTALT
CENTAZ	OBJCTAZ
XBINNING	BINX
YBINNING	BINY
XPIXSZ	XPIXELSZ, PIXSIZE1, PIXSIZEX, XPIXSIZE
YPIXSZ	YPIXELSZ, PIXSIZE2, PIXSIZEY, YPIXSIZE
CCD-TEMP	CCD_TEMP, CCDTEMP, TEMPERAT, CAMTCCD
OFFSET	BLKLEVEL
CVF	EGAIN
FOCPOS	FOCUSPOS
FOCTEMP	FOCUSTEM
STACKCNT	NCOMBINE
SITELAT	SITE-LAT, OBSLAT
SITELONG	SITE-LON, OBSLONG

CHAPITRE 23

Live stacking (empilement en direct)

L'empilement en direct est une technique d'astrophotographie qui permet d'empiler en temps réel une série d'images pour produire une image de meilleure qualité. Contrairement à l'empilement d'images traditionnel, qui consiste à combiner plusieurs images après qu'elles aient été capturées, l'empilement en direct combine les images au moment où elles sont capturées. Cela peut fournir un aperçu instantané de l'image finale et permettre à l'astrophotographe d'effectuer des ajustements en temps réel pour améliorer la qualité du résultat final.

Siril 1.2.0 inclut cette fonctionnalité, qui reste cependant **expérimentale** pour le moment. Elle permet de surveiller un dossier en temps réel et d'empiler les images qui arrivent au fur et à mesure. L'empilement des images peut se faire avec et sans Dark/Bias/Flats. Ces derniers doivent être faits au préalable si vous voulez les utiliser.

23.1 Livestacking (GUI)

> **Note**
>
> Seul les images FITS et RAW sont compatible avec l'empilement en direct.

Pour commencer l'empilement en direct vous devez :

— Définir un répertoire de travail, avec le bouton qui définit le répertoire de travail, dans lequel les photos arriveront les unes après les autres.

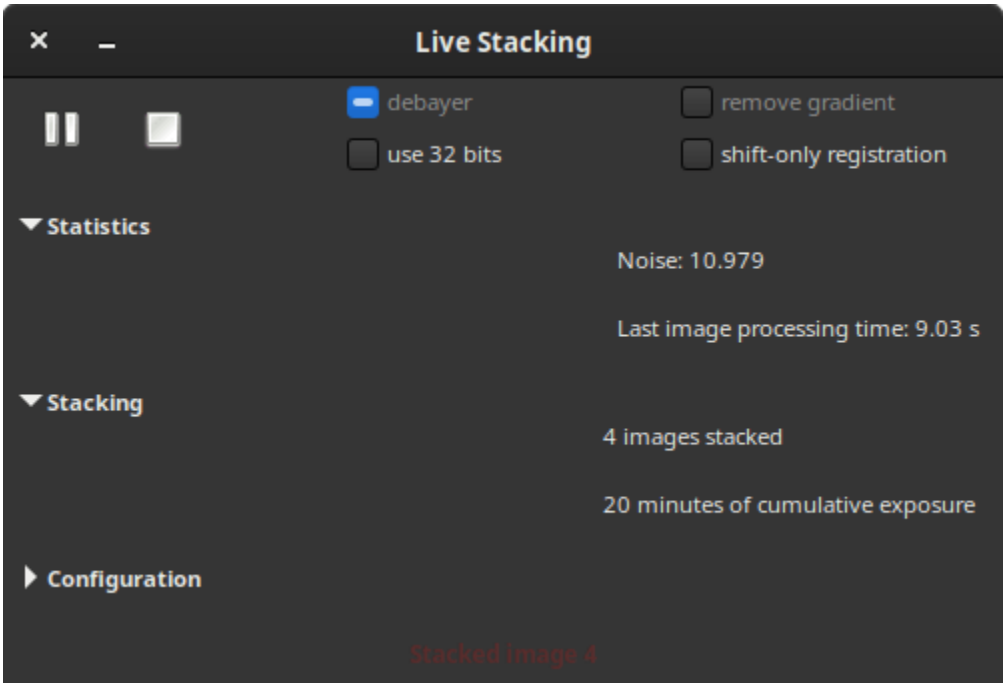
— Cliquer sur le bouton encadré sur l'image ci-dessous.

Une nouvelles fenêtre s'ouvre.

Cette fenêtre contient plusieurs boutons et options. Un bouton play, qui lorsqu'il est cliqué devient un bouton pause, et un bouton stop. Le premier permet de démarrer ou de mettre en pause la surveillance du répertoire de travail, et le dernier de l'arrêter.

Toutes les autres options sont assez standard dans le prétraitement des images astronomiques :

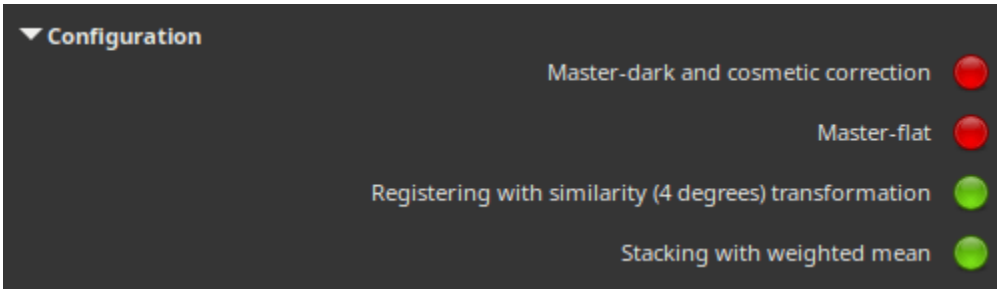
— **dématriçage** : La matrice de Bayer est détectée dans les fichier et le dématricage est automatiquement activé. Il s'agit plus d'un indicateur que d'une option.



— **utiliser 32-bits** : Utiliser 32 bits pour le traitement d'images. Cette méthode est plus lente et généralement inutile en termes de qualité pour l'empilement en direct.
— **supprimer le gradient** : Applique une suppression linéaire du gradient de fond de ciel sur les images d'entrée calibrées.
— **alignement par translation seulement** : Décale uniquement les par translation au lieu d'utiliser la rotation à l'alignement. Cette option doit être décoché pour les montures alt-az ou à forte dérive. Cela rend le traitement des images beaucoup plus rapide.

Les 3 sections ci-dessous fournissent les informations nécessaires à l'utilisateur pour suivre l'évolution de l'empilement.

— **Statistiques** : Cette session donne l'évolution du niveau de bruit en ADU ainsi que le temps de traitement des images.
— **Empilement** : Cette section résume le nombre d'image empilé et le temps total d'exposition cumulé.
— **Configuration** est une section qui n'est pas développée par défaut. Une fois que c'est fait, la section donne des détails si le prétraitement est fait en utilisant des fichiers maîtres et le type d'alignement et d'empilement.



Note

Pour utiliser les fichier maîtres durant une session d'empilement en direct, vous devez d'abord avoir empiler vos fichier. Ainsi, lorsque cela est fait et avant de lancer la session, veuillez les charger dans l'onglet *Calibration*. Ils seront pris en charge lors de l'empilement de direct et seront visible dans la partie *Configuration* de la boite de dialogue live Stacking.

23.2 Empilement en direct (Mode console)

Il est possible d'utiliser l'empilement en direct depuis la ligne de commande. Pour cela, 3 commandes sont nécessaire et expliquées ci-dessous.

— La première, *start_ls* commence la session d'empilement en direct. Il est possible de donner les images darks et flats comme argument pour calibrer les images durant l'empilement en direct.

Ligne de commande Siril

```
start_ls [-dark=filename] [-flat=filename] [-rotate] [-32bits]
```

Initialise une session de livestacking, en utilisant les fichiers de calibration optionnels et attend que les fichiers d'entrée soient fournis par la commande LIVESTACK jusqu'à ce que STOP_LS soit appelé. Le traitement par défaut utilise l'alignement par translation seulement et le traitement 16 bits parce qu'il est plus rapide, il peut être changé en rotation avec **-rotate** et **-32bits**

Notez que les commandes de live stacking mettent Siril dans un état dans lequel il n'est pas capable de traiter d'autres commandes. Après START_LS, seuls LIVESTACK, STOP_LS et EXIT peuvent être appelés jusqu'à ce que STOP_LS soit appelé pour remettre Siril dans son état normal, sans livestacking

Liens : *livestack*, *stop_ls*, *exit*

— La commande *livestack* doit être appliquée à chaque image que vous souhaitez empiler.

Ligne de commande Siril

```
livestack filename
```

Traite l'image fournie pour le livestacking. Cette opération n'est possible qu'après START_LS. Le processus implique la calibration du fichier entrant s'il est configuré dans START_LS, le dématriçage s'il s'agit d'une image couleur, l'alignement et l'empilement. Le résultat temporaire sera dans le fichier live_stack_00001.fit jusqu'à ce qu'une nouvelle option permettant de le modifier soit ajoutée

Liens : *start_ls*

— Enfin, la commande *stop_ls* arrête la session d'empilement en direct.

Ligne de commande Siril

```
stop_ls
```

Arrête la session de livestacking. Uniquement possible après START_LS

Liens : *start_ls*

CHAPITRE 24

Commandes

Cette page liste toutes les commandes disponibles dans Siril 1.4.4.

Vous pouvez accéder à l'index en cliquant sur l'icône .

Les commandes marquées avec l'icône peuvent être utilisées dans des scripts tandis que celles marquées avec celle-ci ne le peuvent pas.

Astuce

Pour toutes les commandes de séquence appelées depuis la barre de commande, vous pouvez remplacer l'argument **sequencename** avec `.` si la séquence à traiter a déjà été chargée.

Astuce

Si vous voulez fournir un argument qui inclut une chaîne de caractères avec des espaces, par exemple un nom de fichier, vous devez citer l'argument entier et pas seulement la chaîne de caractères. Ainsi, par exemple, vous devez utiliser `command "-filename=My File.fits"`, **et non** `command -filename="My File.fits"`.

addmax

```
addmax filename
```

Calcule une nouvelle image en combinant l'image en mémoire avec l'image **filename**. Pour chaque pixel, la nouvelle valeur est déterminée comme étant la valeur maximale dans l'image courante et dans **filename**

asinh

```
asinh [-human] stretch { [offset] [-clipmode=] }
```

Etire l'image pour montrer les objets peu lumineux en utilisant une transformation hyperbolique arcsin. L'argument obligatoire **stretch**, généralement compris entre 1 et 1000, indique l'intensité de l'étirement. Le point noir peut être décalé en fournissant un argument optionnel **offset** dans la valeur normalisée du pixel [0, 1]. Enfin, l'option **-human** permet d'utiliser les facteurs d'efficacité lumineuse de l'œil humain pour déterminer la luminance utilisée pour calculer la valeur d'étirement pour chaque pixel, au lieu de la simple moyenne des valeurs de pixel des canaux. Cette méthode d'étirement préserve la luminosité de l'espace colorimétrique La*b*. Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

autoghs

```
autoghs [-linked] shadowsclip stretchamount [-b=] [-hp=] [-lp=] [-clipmode=]
```

Application de l'étirement hyperbolique généralisé avec un point de symétrie SP défini comme k.sigma de la médiane de chaque canal (la valeur fournie **shadowsclip** est le k ici et peut être négative). Par défaut, SP et l'étirement sont calculés par canal ; SP peut être calculé comme une moyenne des canaux de l'image en passant **-linked**. La quantité d'étirement **D** est fournie dans le deuxième argument obligatoire.

Des valeurs implicites de 13 pour **B**, le rendant très concentré sur la gamme de luminosité SP, 0.7 pour **HP**, 0 pour **LP** sont utilisées, mais peuvent être modifiées avec les options des mêmes noms. Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

autostretch

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

Etire automatiquement l'image actuellement chargée, avec des paramètres différents pour chaque canal (non lié) à moins que **-linked** ne soit fourni. Les arguments sont optionnels, **shadowclip** est le point d'écrêtage des ombres, mesuré en unités sigma à partir du pic de l'histogramme principal (la valeur par défaut est -2.8), **targetbg** est la valeur cible de l'arrière-plan, donnant une luminosité finale à l'image, plage [0, 1], la valeur par défaut est 0.25. Les valeurs par défaut sont celles utilisées dans le rendu Auto-Ajustement à partir de l'interface graphique.

N'utilisez pas la version non liée après l'étalonnage des couleurs, car elle modifierait la balance des blancs

bg

```
bg
```

Retourne le niveau d'arrière-plan de l'image chargée en mémoire

bgnoise

```
bgnoise
```

Renvoie le niveau du bruit de fond de l'image chargée en mémoire

Pour plus d'informations, voir la documentation des *statistiques*

binxy

```
binxy coefficient [-sum]
```

Calcule le binning numérique de l'image en mémoire (somme des pixels 2x2, 3x3..., comme le binning analogique des caméras CCD). Si l'argument facultatif **-sum** est passé, alors la somme des pixels est calculée, tandis que c'est la moyenne lorsque aucun argument facultatif n'est fourni

boxselect

```
boxselect [-clear] [x y width height]
```

Créer une zone de sélection dans l'image actuellement chargée avec les arguments ** x**, **y**, **width** et ** height**, où **x** et **y** sont les coordonnées du coin supérieur gauche commençant à (0, 0), et **width** et ** height**, la taille de la sélection. L'argument **-clear** supprime toute zone de sélection. Si aucun argument n'est transmis, la sélection actuelle est affichée

calibrate

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

Calibre la séquence **sequencename** en utilisant les bias, dark et flat donnés en argument.

Pour les bias, un niveau uniforme peux être spécifié à la place d'une image, en entrant une expression entre guillemet commençant par le signe = tel que -bias="=256" ou -bias="=64*$OFSFSET".

Par défaut, la correction cosmétique n'est pas activée. Si vous voulez en appliquer une, vous devez le spécifier avec l'option **-cc=**.
Vous pouvez utiliser l'option **-cc=dark** pour détecter les pixels chauds et froids à partir du master dark (un master dark doit être fourni avec l'option **-dark=**), éventuellement suivi de **siglo** et **sighi** pour les pixels froids et chauds respectivement. Une valeur de 0 désactive la correction. Si les sigmas ne sont pas fournis, seule la détection des pixels chauds avec un sigma de 3 sera appliquée.
Alternativement, vous pouvez utiliser l'option **-cc=bpm** suivie du chemin vers votre carte de mauvais pixels pour spécifier quels pixels doivent être corrigés. Un fichier d'exemple peut être obtenu avec la commande *find_hot* sur un master dark.

Trois options s'appliquent aux images couleur (au format CFA) : **-cfa** pour les corrections cosmétiques, **-debayer** pour dématricer les images avant de les enregistrer, et **-equalize_cfa** pour égaliser l'intensité moyenne des couches RVB du flat maître, afin d'éviter de teinter l'image calibrée.
L'option **-fix_xtrans** est dédiée aux fichiers X-Trans en appliquant une correction sur les Darks et les Biais/offset pour supprimer un vilain motif carrée.
Il est également possible d'optimiser la soustraction des darks avec **-opt**, qui nécessite la fourniture de masters de bias et de dark, et calcule automatiquement le coefficient à appliquer au dark, ou calcule le coefficient grâce au mot-clé "exposure" avec **-opt=exp**.
Par défaut, les images marquées comme exclues ne sont pas traitées. L'argument **-all** peut être utilisé pour forcer le traitement de toutes les images même si elles sont marquées comme exclues.
Le nom de la séquence de sortie comme par le préfixe "pp_" sauf indication contraire spécifié avec l'option **-prefix=**.
Si **-fitseq** est fourni, la séquence de sortie sera une séquence FITS (fichier unique)

calibrate_single

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

Calibre l'image **imagename** en utilisant les bias, dark et flat donnés en argument.

Pour les bias, un niveau uniforme peux être spécifié à la place d'une image, en entrant une expression entre guillemet commençant par le signe = tel que -bias="=256" ou -bias="=64*$OFSFSET".

Par défaut, la correction cosmétique n'est pas activée. Si vous voulez en appliquer une, vous devez le spécifier avec l'option **-cc=**.

Vous pouvez utiliser l'option **-cc=dark** pour détecter les pixels chauds et froids à partir du master dark (un master dark doit être fourni avec l'option **-dark=**), éventuellement suivi de **siglo** et **sighi** pour les pixels froids et chauds respectivement. Une valeur de 0 désactive la correction. Si les sigmas ne sont pas fournis, seule la détection des pixels chauds avec un sigma de 3 sera appliquée.

Alternativement, vous pouvez utiliser l'option **-cc=bpm** suivie du chemin vers votre carte de mauvais pixels pour spécifier quels pixels doivent être corrigés. Un fichier d'exemple peut être obtenu avec la commande *find_hot* sur un master dark.

Trois options s'appliquent aux images couleur (au format CFA) : **-cfa** pour les corrections cosmétiques, **-debayer** pour dématricer les images avant de les enregistrer, et **-equalize_cfa** pour égaliser l'intensité moyenne des couches RVB du flat maître, afin d'éviter de teinter l'image calibrée.

L'option **-fix_xtrans** est dédiée aux fichiers X-Trans en appliquant une correction sur les Darks et les Biais/offset pour supprimer un vilain motif carrée.

Il est également possible d'optimiser la soustraction des darks avec **-opt**, qui nécessite la fourniture de masters de bias et de dark, et calcule automatiquement le coefficient à appliquer au dark, ou calcule le coefficient grâce au mot-clé "exposure" avec **-opt=exp**

Le nom du fichier de sortie comme par le préfixe "pp_" sauf indication contraire spécifié avec l'option **-prefix=**

capabilities

```
capabilities
```

Affiche les fonctionnalités de Siril, en fonction de la compilation et de l'exécution

catsearch

```
catsearch name
```

Recherche un objet par **nom** et l'ajoute au catalogue d'annotations de l'utilisateur. L'objet est d'abord recherché dans les catalogues d'annotations, s'il n'est pas trouvé, une demande est faite à SIMBAD.

L'objet peut être un objet du système solaire, auquel cas un préfixe, "a :" pour astéroïde, "p :" pour planète, "c :" pour comète, "dp :" pour planète naine ou "s :" pour satellite naturel, doit précéder le nom de l'objet. La recherche est effectuée pour la date, l'heure et le lieu d'observation figurant dans l'en-tête de l'image, à l'aide du service Miriade de l'IMCCE

ccm

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

Applique une matrice de conversion des couleurs à l'image en cours.

Il y a 9 arguments obligatoires correspondant aux 9 éléments de la matrice :

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

Un dixième argument supplémentaire **[gamma]** peut être fourni : s'il est omis, la valeur par défaut est 1,0.

Ceux-ci sont appliqués à chaque pixel selon les formules suivantes :

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

cd

```
cd directory
```

Définit le nouveau répertoire de travail courant.

L'argument **directory** peut contenir le jeton ~, qui est remplacé par le répertoire personnel, les répertoires comportant des espaces dans leur nom peuvent être protégés en utilisant des guillemets simples ou doubles

cdg

```
cdg
```

Renvoie les coordonnées du centre de gravité de l'image. Seuls les pixels dont la valeur est supérieure à 15,7% de l'ADU maximale et dont les quatre voisins remplissent les mêmes conditions sont utilisés pour le calculer, et il n'est calculé que s'il y en a au moins 50

clahe

```
clahe cliplimit tileSize
```

Égalise l'histogramme d'une image en utilisant l'égalisation d'histogramme par contraste adaptatif limité.

cliplimit définit le seuil pour la limitation du contraste.
tilesize définit la taille de la grille pour l'égalisation d'histogramme. L'image d'entrée sera divisée en tuiles rectangulaires de taille égale

clear

```
clear
```

Efface les logs de la console

clearstar

```
clearstar
```

Efface toutes les étoiles sauvegardées en mémoire et affichées à l'écran

close

```
close
```

Ferme correctement l'image ou la séquence ouverte, le cas échéant

conesearch

```
conesearch [limit_magnitude] [-cat=] [-phot] [-obscode=] [-tag={on|off}] [-log={on|off}]␣
↪[-trix=] [-out=]
```

Affiche les étoiles du catalogue local par défaut pour l'image chargée résolue, jusqu'à la magnitude limite **limit_magnitude** fournie (13 par défaut pour la plupart des catalogues, sauf 14.5 pour aavso_chart, 20 pour solsys, et omis pour pgc).

Un autre catalogue en ligne peut être spécifié avec **-cat=**, prenant les valeurs

- for stars : tycho2, nomad, gaia, localgaia, ppmxl, bsc, apass, gcvs, vsx, simbad, aavso_chart
- for exoplanets : exo
- for deep-sky : pgc
- for solar system objects : solsys (closest IAU observatory code can be passed with the argument **-obscode=** for better position accuracy)

Pour les catalogues d'étoiles contenant des données photométriques, les étoiles sans information B-V seront conservées ; elles peuvent être exclues en passant l'option **-phot**

L'argument **-trix=** peut être passé à la place d'un catalogue suivi d'un nombre entre 0 et 511 pour tracer les étoiles contenues dans les catalogues locaux trixel de niveau 3 (pour l'usage des développeurs principalement)

Certains catalogues (bsc, gcvs, pgc, exo, aavso_chart, varisum et solsys) affichent également, par défaut, les noms à côté des marqueurs dans l'écran (interface graphique uniquement) et les listent dans le journal. Pour d'autres qui ont un plus grand nombre d'objets, à savoir vsx et simbad, l'information peut aussi être affichée, mais, comme elle risque d'encombrer l'écran, elle n'est pas activée par défaut. Ce comportement peut être activé/désactivé avec les options **-tag=on|off** pour afficher les noms à côté des marqueurs et **-log=on|off** pour lister les objets dans le journal de la console

La liste des éléments présents dans l'image peut éventuellement être sauvegardée dans un fichier csv en passant l'argument **-out=**

convert

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

Convertit toutes les images du répertoire de travail actuel qui sont dans un format supporté en séquence d'images FITS de Siril (plusieurs fichiers) ou en séquence FITS (un seul fichier) si **-fitseq** est fourni ou en séquence SER (un seul fichier) si **-ser** est fourni. L'argument **basename** est le nom de base de la nouvelle séquence, les nombres et l'extension seront mis derrière.

Pour les images FITS, Siril essaiera de faire un lien symbolique ; si ce n'est pas possible, les fichiers seront copiés. L'option **-debayer** applique le dématriçage aux images d'entrée CFA ; dans ce cas, aucun lien symbolique n'est effectué.

-start=index définit le numéro d'index de départ, utile pour continuer une séquence existante (non utilisé avec -fitseq ou **-ser** ; assurez-vous de supprimer ou d'effacer le .seq cible s'il existe dans ce cas).

L'option **-out=** modifie le répertoire de sortie en fonction de l'argument fourni.

Voir aussi CONVERTRAW et LINK

Liens : *convertraw*, *link*

convertraw

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

Identique à CONVERT mais ne convertit que les fichiers RAW des reflex numériques se trouvant dans le répertoire de travail actuel

Liens :*convert*

cosme

```
cosme [filename].lst
```

Applique la moyenne locale à un ensemble de pixels de l'image chargée (correction cosmétique). Les coordonnées de ces pixels se trouvent dans un fichier texte [.lst file], la commande FIND_HOT peut également le créer pour des pixels chauds uniques, mais une opération manuelle est nécessaire pour supprimer des lignes ou des colonnes. COSME est adapté pour corriger les pixels chauds et froids résiduels après la calibration.

Au lieu de fournir la liste des mauvais pixels, il est également possible de les détecter dans l'image courante à l'aide de la commande FIND_COSME

Liens : *find_hot*, *find_cosme*

Format de fichier pour la liste des mauvais pixels : * Les lignes de la forme *P x y* corrigent le pixel aux coordonnées (x, y), type est un caractère optionnel (C ou H) spécifiant à Siril si le mauvais pixel courant est froid ou chaud. Cette ligne est créée par la commande FIND_HOT mais vous pouvez aussi ajouter les deux types de lignes suivants manuellement * Les lignes de la forme *C x 0 `corrigeront la mauvaise colonne aux coordonnées x. * Les lignes de la forme `L y 0* fixeront la mauvaise ligne aux coordonnées y.

cosme_cfa

```
cosme_cfa [filename].lst
```

Même fonction que COSME mais qui s'applique aux images RAW CFA

Liens :*cosme*

crop

```
crop [x y width height]
```

Recadre dans une zone sélectionnée de l'image chargée.

Si une sélection est active, aucun autre argument n'est nécessaire. Sinon, ou dans les scripts, les arguments doivent être donnés, avec **x** et **y** étant les coordonnées du coin supérieur gauche, et **width** et **height** la taille de la sélection. Alternativement, la sélection peut être effectuée à l'aide de la commande BOXSELECT

Liens : *boxselect*

ddp

```
ddp level coef sigma
```

Effectue un traitement de développement numérique (DDP) sur l'image chargée, comme l'a d'abord décrit Kunihiko Okano. Cette implémentation est celle décrite dans IRIS.

Il combine une distribution linéaire sur les niveaux de luminosité faibles (inférieurs à **level**) et une non linéaire sur les niveaux élevés.
Il utilise un filtre Gaussien de l'écart-type **sigma** et multiplie l'image résultante par **coef**. Les valeurs typiques pour **sigma** se situent entre 0,7 et 2. L'argument niveau doit être compris dans la plage [0, 65535] pour les images 16 bits et peut être donné dans la plage [0, 1] ou [0, 65535] pour les images 32 bits, auquel cas il sera automatiquement mis à l'échelle

denoise

```
denoise [-nocosmetic] [-mod=m] [ -vst | -da3d | -sos=n [-rho=r] ] [-indep]
```

Réduit le bruit de l'image en utilisant l'algorithme bayésien non local décrit par Lebrun, Buades et Morel.

Il est fortement recommandé d'appliquer une correction cosmétique pour éliminer le bruit sel et poivre avant d'exécuter le réducteur de bruit, et par défaut cette commande applique automatiquement la correction cosmétique. Toutefois, si celle-ci a déjà été effectuée plus tôt dans le flux de travail, elle peut être désactivée ici à l'aide de la commande optionnelle **-nocosmetic**.

Un argument optionnel **-mod=m** peut être donné, où 0 <= m <= 1. Le pixel de sortie est calculé comme suit : *out=m x d + (1 m) x in*, où *d* est la valeur de pixel réduite du bruit. Une valeur de modulation de 1 n'appliquera aucune modulation. Si le paramètre est omis, il est par défaut à 1.

L'argument optionnel **-vst** peut être utilisé pour appliquer la transformation de stabilisation de variance Anscombe généralisée avant NL-Bayes. Cela est utile avec des images à faible luminosité telles que des images simples, où le bruit suit une distribution de Poisson ou Poisson-Gaussien plutôt qu'une distribution principalement gaussienne. Il ne peut pas être utilisé en conjonction avec DA3D ou SOS, et pour le débruitage d'images empilées, il n'est généralement pas bénéfique.

L'argument optionnel **-da3d** peut être utilisé pour activer le débruitage Data-Adaptive Dual Domain (DA3D) en tant qu'algorithme de débruitage final. Cela utilise la sortie de BM3D comme image de guidage pour affiner le débruitage. Il améliore les détails et réduit les artefacts en escalier.

L'argument optionnel **-sos=n** peut être utilisé pour activer le renforcement itératif de la détection de bruit Strengthen-Operate-Subtract (SOS), avec le nombre d'itérations spécifié par n. En particulier, ce renforceur peut produire de meilleurs résultats si l'algorithme NL-Bayes non renforcé produit des artefacts dans les zones de fond de ciel. Si à la fois -da3d et -sos=n sont spécifiés, le dernier spécifié sera appliqué.

L'argument facultatif **-rho=r** peut être spécifié, où 0 < r < 1. Cela est utilisé par le booster SOS pour déterminer la quantité d'image bruitée ajoutée au résultat intermédiaire entre chaque itération. Si -sos=n n'est pas spécifié, le paramètre est ignoré.

Par défaut, DA3D ou SOS ne sont pas appliqués, car l'amélioration de la réduction de bruit est généralement relativement faible et ces techniques nécessitent un temps de traitement supplémentaire.

En de très rares occasions ,lors du débruitage des images en couleur, des artefacts de couleur en bloc peuvent apparaître. L'argument optionnel **-indep** peut être utilisé pour éviter cela en effectuant le débruitage de chaque canal séparément. Cela est plus lent mais éliminera les artefacts

dir

```
dir
```

Liste des fichiers et répertoires dans le répertoire de travail

Cette commande est disponible uniquement sur Microsoft Windows, pour une commande équivalente sur Linux et MacOS, voir *ls*.

disto

```
disto [clear]
```

Montre le champ de distorsion sur une image résolue dont la solution comprend des termes de distorsion

Passer l'option **clear** pour désactiver

dumpheader

```
dumpheader
```

Affiche l'en-tête FITS de l'image chargée dans la console

entropy

```
entropy
```

Calcule l'entropie de l'image chargée sur la couche affichée, uniquement dans la zone sélectionnée si une zone a été sélectionnée ou dans l'image entière. L'entropie est une façon de mesurer le bruit ou les détails d'une image

epf

```
epf [-guided] [-d=] [-si=] [-ss=] [-mod=] [-guideimage=]
```

Applique un filtre préservant les bords. Par défaut, un filtre bilatéral est appliqué ; un filtre guidé peut être spécifié en utilisant l'argument **-guided**. Le diamètre du filtre est fixé par défaut à 3 et peut être défini à l'aide de **-d=**. Attention aux valeurs de d supérieures à 20, car l'algorithme peut être coûteux en termes de calcul.

La valeur sigma du filtrage d'intensité peut être définie à l'aide de **-si=** et la valeur sigma spatiale peut être définie à l'aide de **-ss=**. Les valeurs sigma représentent la différence entre les valeurs des pixels sur lesquelles le filtre agit fortement : pour les images 32 bits, la valeur doit être comprise entre 0 et 1,0, tandis que pour les images 16 bits, elle

doit être comprise entre 0 et 65535. Par défaut, si rien n'est spécifié, les deux valeurs sont fixées à 11. Si **-d=0** est défini, le diamètre du filtre sera défini automatiquement en fonction de la valeur de **-ss**. *Notez que lors de l'application d'un filtre guidé, seul *-sc** s'applique.*

Lors de la spécification d'un filtre guidé, une image guide peut être définie en utilisant **-guideimage=**. Par défaut, si aucune image guide n'est spécifiée, un filtre auto-guidé est effectué. *Note : l'image guide doit avoir les mêmes dimensions que l'image à filtrer !*

L'intensité du filtre peut être modulée à l'aide de l'argument **-mod=**. Si mod = 1.0, l'effet complet du filtre sera appliqué ; pour mod inférieur à 1.0, une proportion de l'image originale sera mélangée au résultat, et pour mod = 0.0, aucun filtrage ne sera appliqué

exit

```
exit
```

Quitte l'application

extract

```
extract NbPlans
```

Extrait les plans **NbPlans** du domaine des ondelettes de l'image chargée.
Voir aussi WAVELET et WRECONS. Pour l'extraction des couleurs, voir SPLIT

Liens : *wavelet*, *wrecons*, *split*

extract_Green

```
extract_Green
```

Extrait le signal vert de l'image CFA chargée. Siril lit les informations de la matrice de Bayer à partir de l'image ou des préférences et exporte uniquement les données moyennes du filtre vert dans un nouveau fichier FITS de taille réduite. Un nouveau fichier est créé, son nom est préfixé par "Green_"

extract_Ha

```
extract_Ha [-upscale]
```

Extrait le signal H-Alpha de l'image CFA chargée. Il lit les informations de la matrice de Bayer à partir de l'image ou des préférences et exporte uniquement les données du filtre rouge dans un nouveau fichier FITS de taille réduite. Si l'argument **-upscale** est fourni, la sortie sera mise à l'échelle x2 pour correspondre à la résolution complète du capteur, par exemple pour correspondre à d'autres images produites par la même famille de capteurs. Un nouveau fichier est créé, son nom est préfixé par "Ha_"

extract_HaOIII

```
extract_HaOIII [-resample=]
```

Extrait les signaux H-Alpha et O-III de l'image CFA chargée. Siril lit les informations de la matrice de Bayer à partir de l'image ou des préférences et exporte uniquement les données du filtre rouge pour H-Alpha dans un nouveau fichier FITS de taille réduite (comme EXTRACTHA) et conserve les trois autres pour O-III avec un remplacement interpolé pour le pixel rouge. Les noms des fichiers de sortie commencent par le préfixe "Ha_" et "OIII_"

L'argument optionnel **-resample={ha|oiii}** indique s'il faut suréchantillonner l'image Ha ou sous-échantillonner l'image OIII pour avoir des images de même taille. Si cet argument n'est pas fourni, aucun rééchantillonnage ne sera effectué et l'image OIII aura deux fois la hauteur et la largeur de l'image Ha

fdiv

```
fdiv filename scalar
```

Divise l'image chargée par l'image donnée en argument. L'image résultante est multipliée par la valeur de l'argument **scalar**. Voir aussi IDIV

Liens : *idiv*

ffill

```
ffill value [x y width height]
```

Même commande que FILL mais il s'agit d'un remplissage symétrique d'une région définie par la souris ou avec BOXSELECT. Utilisé pour traiter une image dans le domaine de Fourier (FFT)

Liens : *fill*, *boxselect*

fftd

```
fftd modulus phase
```

Applique une transformée de Fourier rapide à l'image chargée. **modulus** et **phase** donnés en argument sont les noms des fichiers FITS sauvegardés

ffti

```
ffti modulus phase
```

Récupère l'image corrigée en appliquant une transformation inverse. Les arguments **modulus** et **phase** sont les noms des fichiers d'entrée, le résultat sera la nouvelle image chargée

fill

```
fill value [x y width height]
```

Remplit l'image chargée entièrement ou seulement la sélection s'il y en a une avec des pixels ayant l'intensité **valeur** exprimée en ADU

find_cosme

```
find_cosme cold_sigma hot_sigma
```

Détection et remplacement automatique des pixels froids et chauds dans l'image chargée, avec les seuils transmis en arguments en unités sigma

find_cosme_cfa

```
find_cosme_cfa cold_sigma hot_sigma
```

Même commande que FIND_COSME mais pour les images CFA

Liens : *find_cosme*

find_hot

```
find_hot filename cold_sigma hot_sigma
```

Enregistre dans le répertoire de travail un fichier liste **filename** (format texte) contenant les coordonnées des pixels dont l'intensité **hot_sigma** fois supérieure et **cold_sigma** fois inférieure à l'écart-type, extraits de l'image chargée. Nous utilisons généralement cette commande sur un fichier dark maître. La commande COSME permet d'appliquer cette liste de mauvais pixels à une image chargée, voir aussi SEQCOSME pour l'appliquer à une séquence

Liens : *cosme*, *seqcosme*

Les lignes `P x y type` vont fixer le pixel aux coordonnées (x, y), où "type" est un caractère optionnel (C ou H) spécifiant à Siril si le pixel actuel est froid ou chaud. Cette ligne est créée par la commande FIND_HOT, mais vous pouvez également ajouter manuellement quelques lignes :
Les lignes `C x 0 type` vont fixer la colonne défectueuse aux coordonnées x.
Les lignes `L y 0 type` vont fixer la ligne défectueuse aux coordonnées y.

findcompstars

```
findcompstars star_name [-narrow|-wide] [-catalog={nomad|apass}] [-dvmag=3] [-dbv=0.5] [-
↪emag=0.03] [-out=nina_file.csv]
```

Recherche automatiquement des étoiles de comparaison dans le champ de l'image chargée résolue en astrométrie, pour l'analyse photométrique de la courbe de lumière d'une étoile selon les critères suivants
- the provided name of the star
- the field of view of the image, reduced to a diameter of its height if **-narrow** is passed, avoiding stars in the corners
- the chosen catalog (APASS by default), can be changed with **-catalog={NOMAD|APASS}**
- the difference in visual magnitude from the variable star, in the range [0, 6] with a default of 3, changed with **-dvmag=**
- the difference in color with the variable star, in the range [0.0, 0.7] of their B-V indices with a default of 0.5, changed with **-dbv=**
- the maximum allowed error on Vmag in the range [0.0, 0.1] with a default of 0.03, changed with **-emag=**.

La liste peut éventuellement être sauvegardée sous la forme d'un fichier CSV compatible avec la liste des étoiles de comparaison de NINA, en spécifiant le nom du fichier avec **-out=**. Si le nom fourni est la valeur spéciale **auto**, il est généré en utilisant les paramètres d'entrée

Voir aussi LIGHT_CURVE

Links : *light_curve*

findstar

```
findstar [-out=] [-layer=] [-maxstars=]
```

Détecte les étoiles dans l'image actuellement chargée, dont le niveau est supérieur à un seuil calculé par Siril.
Ensuite, une PSF est appliquée et Siril rejette toutes les structures détectées qui ne répondent pas à un ensemble de critères de détection prescrits, qui peuvent être ajustés avec la commande SETFINDSTAR.
Enfin, une ellipse est dessinée autour des étoiles détectées.

Le paramètre optionnel **-out=** permet d'enregistrer les résultats dans le chemin d'accès donné.
L'option **-layer=** spécifie la couche sur laquelle la détection est effectuée (pour les images en couleur uniquement).
Vous pouvez également limiter le nombre maximum d'étoiles détectées en passant une valeur à l'option **-maxstars=**.

Voir aussi la commande CLEARSTAR

Liens : *psf*, *setfindstar*, *clearstar*

fix_xtrans

```
fix_xtrans
```

Corrige les pixels Fujifilm X-Trans Auto Focus dans l'image chargée.

En effet, en raison du système de mise au point automatique à détection de phase, les photosites utilisés pour la mise au point automatique reçoivent un peu moins de lumière que les photosites environnants. L'appareil photo compense cela et augmente les valeurs de ces photosites spécifiques, ce qui donne un carré visible au milieu des images dark/bias

fixbanding

```
fixbanding amount sigma [-vertical]
```

Tente de supprimer les trames horizontales ou verticales de l'image chargée.
amount définit le montant de la correction, entre 0 et 4.
sigma définit le niveau de protection de l'algorithme, plus sigma est élevé, plus la protection est importante, entre 0 et 5. Des valeurs de 1 et 1 sont souvent suffisantes.
L'option **-vertical** permet de supprimer les trames verticales, l'option horizontale étant la valeur par défaut

fmedian

```
fmedian ksize modulation
```

Effectue un filtre médian de taille **ksize** x **ksize** (**ksize** DOIT être impair) sur l'image chargée avec un paramètre de modulation **modulation**.

Le pixel de sortie est calculé comme suit : out = mod x m + (1 mod) x in, où m est la valeur du pixel filtré par médiane. Une valeur de modulation de 1 n'appliquera aucune modulation

fmul

```
fmul scalar
```

Multiplie l'image chargée par le scalaire **scalar** donné en argument

gauss

```
gauss sigma
```

Applique à l'image chargée un flou gaussien avec le **sigma** donné.

Voir aussi UNSHARP, la même chose avec un paramètre de mélange

Liens : *unsharp*

get

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

Obtient une valeur des paramètres en utilisant son nom, ou liste tout avec **-a** (liste de noms et de valeurs) ou avec **-A** (liste détaillée)

Voir aussi SET pour mettre à jour les valeurs

Liens : *set*

getref

```
getref sequencename
```

Affiche des informations sur l'image de référence de la séquence donnée en argument. La première image a l'index 0

ght

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

Étirement hyperbolique généralisé basé sur le travail de l'équipe ghsastro.co.uk.

L'argument **-D=** définit la force de l'étirement, entre 0 et 10. C'est le seul argument obligatoire. Les arguments facultatifs suivants permettent de mieux adapter l'étirement :

B définit l'intensité de l'étirement près du point focal, entre -5 et 15 ;

LP définit une plage de préservation des ombres entre 0 et SP où l'étirement sera linéaire, préservant les détails d'ombre ;

SP définit le point de symétrie de l'étirement, entre 0 et 1, qui est le point auquel l'étirement est le plus intense ;

HP définit une région entre HP et 1 où l'étirement est linéaire, préservant les détails de la mise en évidence et évitant le gonflement des étoiles.

Si elle est omise, B, LP et SP ont une valeur par défaut de 0.0 et HP a une valeur par défaut de 1.0.

Un argument facultatif (soit **-human**, **-even** ou **-independent**, peut être passé pour sélectionner la luminance pondérée humaine ou pondérée paire ou les canaux de couleur indépendants pour les étirements de couleur). L'argument est ignoré pour les images mono. Alternativement, l'argument **-sat** spécifie que l'étirement est effectué sur la saturation de l'image - l'image doit être en couleur et tous les canaux doivent être sélectionnés pour que cela fonctionne.

En option, le paramètre **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, V, B, RV, RB ou VB. La valeur par défaut est « tous les canaux ». Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

grey_flat

```
grey_flat
```

Égalise l'intensité moyenne des couches RVB dans l'image CFA chargée. Il s'agit du même processus que celui utilisé sur les images flats lors de la calibration, lorsque l'option "equalize CFA" est utilisée

help

```
help [command]
```

Liste des commandes disponibles ou aide pour une commande

histo

```
histo channel (channel=0, 1, 2 with 0: red, 1: green, 2: blue)
```

Calcule l'histogramme du **canal** de l'image chargée et produit le fichier histo_[nom du canal].dat dans le répertoire de travail.
layer = 0, 1 ou 2 avec 0=rouge, 1=vert and 2=bleu

iadd

```
iadd filename
```

Ajoute l'image **filename** à l'image chargée.
Le résultat sera en 32 bits par canal si les préférences le permettent

icc_assign

```
icc_assign profile
```

Affecte le profil ICC spécifié dans l'argument à l'image actuelle.
L'un des arguments spéciaux suivants peut être fourni pour utiliser les profils intégrés respectifs : **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **working** pour définir le profil de couleur de travail monochrome ou RVB, (pour les images monochromes uniquement) **linear**, ou le chemin d'accès à un fichier de profil ICC peut être fourni. Si un profil intégré est spécifié avec une image monochrome chargée, le profil Gray avec la CRT correspondant sera utilisé

icc_convert_to

```
icc_convert_to profile [intent]
```

Convertit l'image actuelle au profil ICC spécifié.
L'un des arguments spéciaux suivants peut être fourni pour utiliser les profils intégrés respectifs : **sRGB**, **sRGBlinear**, **Rec2020**, **Rec2020linear**, **graysrgb**, **grayrec2020**, **graylinear** ou **working** pour définir le profil de travail de couleur monochrome ou RVB, (pour les images monochromes uniquement) **linear**, ou le chemin d'accès à un fichier de profil ICC peut être fourni. Si un profil intégré est spécifié avec une image monochrome chargée, le profil Gray avec la CRT correspondant sera utilisé.

Un deuxième argument peut être fourni pour spécifier l'intention de transformation de la couleur : il doit s'agir d'un des éléments suivants : **perceptual**, **relative** (pour la colorimétrie relative), **saturation** ou **absolute** (pour la colorimétrie absolue)

icc_remove

```
icc_remove
```

Supprime le profil ICC de l'image actuelle, si elle en possède un

idiv

```
idiv filename
```

Divise l'image chargée par l'image **filename**.
Le résultat sera en 32 bits par canal si les préférences le permettent.

Voir aussi FDIV

Liens : *fdiv*

imul

```
imul filename
```

Multiplie l'image **filename** par l'image chargée.
Le résultat sera en 32 bits par canal si les préférences le permettent

inspector

```
inspector
```

Divise l'image chargée en une mosaïque de neuf panneaux montrant les coins et le centre de l'image pour une inspection plus approfondie (interface graphique uniquement)

invght

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

Inverse un étirement hyperbolique généralisé. Il s'agit de la transformation inverse de l'étirement hyperbolique généralisé. S'il est fourni avec les mêmes paramètres, il annule une commande d'étirement hyperbolique généralisé, ce qui permet de revenir à une image linéaire. Il peut également fonctionner de la même manière que le GHT, mais pour des images en négatif

Liens : *ght*

invmodasinh

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

Inverse un étirement arcsinh modifié. Il fournit la transformation inverse de MODASINH et, s'il est fourni avec les mêmes paramètres, il annule une commande MODASINH, ce qui permet de revenir à une image linéaire. Il peut également fonctionner de la même manière que MODASINH, mais pour des images en négatif

Liens : *modasinh*

invmtf

```
invmtf low mid high [channels]
```

Inverse une fonction de transfert de tons moyens. Elle fournit la transformation inverse de la MTF et, si elle est fournie avec les mêmes paramètres, elle annule une commande MTF, ce qui permet de revenir à une image linéaire.

Elle peut également fonctionner de la même manière que la MTF, mais pour les images en négatif

Liens : *mtf*

isub

```
isub filename
```

Soustrait l'image chargée de l'image **filename**.
Le résultat sera en 32 bits par canal si les préférences le permettent, ce qui permet de stocker des valeurs négatives. Pour couper une valeur négative, utilisez le mode 16 bits ou la commande THRESHLO

Liens : *threshlo*

jsonmetadata

```
jsonmetadata FITS_file [-stats_from_loaded] [-nostats] [-out=]
```

Exporte les métadonnées et les statistiques de l'image actuellement chargée sous forme JSON. Le nom de fichier est requis, même si l'image est déjà chargée. Les données de l'image ne peuvent pas être lues à partir du fichier si l'image est déjà chargée et si l'option **-stats_from_loaded** est spécifiée. Les statistiques peuvent être désactivées en fournissant l'option **-nostats**. Un fichier contenant les données JSON est créé avec le nom de fichier par défaut '$(FITS_nom_de_fichier_sans_extension).json' et peut être modifié avec l'option **-out=**

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 plusieurs étoiles avec photométrie d'ouverture dans une séquence d'images et produit une courbe de lumière pour l'une d'entre elles, calibrée par les autres. Les premières coordonnées, en pixels si **-at=** est utilisé ou en degrés si **-wcs=** est utilisé, sont pour l'étoile dont la lumière sera tracée, les autres pour les étoiles de comparaison.
Alternativement, une liste d'étoiles cibles et de référence peut être transmise dans le format de la liste d'étoiles du plugin exoplanète de NINA, avec l'option **-ninastars=**. Siril vérifie que toutes les étoiles de référence peuvent être

utilisées avant de les utiliser. Un fichier de données est créé dans le répertoire courant nommé light_curve.dat, puis Siril trace le résultat dans une image PNG

Les rayons de l'anneau peuvent être configurés dans les paramètres ou fixés à un facteur de la FWHM de l'image de référence si l'option **-autoring** est activée. Ces tailles de rayons sont 4,2 fois et 6,3 fois le FWHM pour les rayons intérieurs et extérieurs, respectivement.

Voir également la commande **setphot** pour définir de la même manière la taille du rayon d'ouverture.

Voir aussi SEQPSF pour les opérations sur une seule étoile

Liens : *seqpsf*

limit

```
limit { -clip | -posrescale | -rescale }
```

Limite les valeurs des pixels dans les images 32 bits à la plage de 0,0 à 1,0. Cette commande ne s'applique pas aux images 16 bits, car il ne peut y avoir de valeurs hors plage. La limitation de la plage peut être effectuée de l'une des manières suivantes :

-clip : cette option écrête simplement tous les pixels négatifs à 0,0 et tous les pixels ayant une valeur > 1,0 à 1,0.

-posrescale : cette option met à l'échelle toutes les valeurs positives des pixels de manière que la valeur maximale soit de 1,0, les pixels négatifs étant ramenés à 0,0. Pour les images à trois canaux, le même facteur d'échelle est appliqué à tous les canaux. Si la valeur maximale du pixel est déjà <= 1,0, les pixels négatifs seront toujours écrêtés, mais aucun facteur d'échelle ne sera appliqué aux pixels positifs.

-rescale : using this option, if there are any negative pixel values the image will have a constant value added to all pixel values so that the minimum value is 0.0. Then if the maximum pixel value is > 1.0, a scaling factor is applied so that the maximum pixel value is scaled to 1.0.

Note that if there are one or more extreme outliers (for example as a result of bad pixels) the **-rescale** and **-posrescale** options may produce an unexpected result. This can be mitigated by applying cosmetic correction to the image first

linear_match

```
linear_match reference low high
```

Calcule et applique une fonction linéaire entre une image de **référence** et l'image chargée.

L'algorithme ignore tous les pixels de référence dont les valeurs se situent en dehors de la plage [**low**, **high**]

link

```
link basename [-date] [-start=index] [-out=]
```

Identique à CONVERT mais ne convertit que les fichiers FITS se trouvant dans le répertoire de travail actuel. Ceci est utile pour éviter les conversions de résultats JPEG ou d'autres fichiers qui peuvent se retrouver dans le répertoire. L'argument supplémentaire **-date** permet de trier les fichiers en fonction de leur valeur DATE-OBS plutôt qu'en fonction de leur nom alphanumérique

Liens :*convert*

linstretch

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

Étire linéairement l'image jusqu'à un nouveau point noir BP.

Le paramètre optionnel **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, G, B, RG, RB ou GB. La valeur par défaut est tous les canaux.

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. The clip mode can be set using the argument **-clipmode=** : values **clip**, **rescale**, **rgbblend** or **globalrescale** are accepted and the default is rgbblend

livestack

```
livestack filename
```

Traite l'image fournie pour le livestacking. Cette opération n'est possible qu'après START_LS. Le processus implique la calibration du fichier entrant s'il est configuré dans START_LS, le dématriçage s'il s'agit d'une image couleur, l'alignement et l'empilement. Le résultat temporaire sera dans le fichier live_stack_00001.fit jusqu'à ce qu'une nouvelle option permettant de le modifier soit ajoutée

Liens : *start_ls*

Avertissement

Notez que les commandes de livestacking mettent Siril dans un état dans lequel il n'est pas capable de traiter d'autres commandes. Après START_LS, seuls LIVESTACK, STOP_LS et EXIT peuvent être appelés jusqu'à ce que STOP_LS soit appelé pour remettre Siril dans son état normal, sans livestacking.

load

```
load filename[.ext]
```

Charge l'image **filename** à partir du répertoire de travail actuel, qui devient l'"image actuellement chargée" utilisée dans de nombreuses commandes d'image unique.

Il tente d'abord de charger **filename**, puis **filename**.fit, **filename**.fits et enfin tous les formats pris en charge.

Ce schéma s'applique à toutes les commandes Siril qui impliquent la lecture de fichiers

load_seq

```
load_seq sequencename[.ext]
```

Loads the sequence **sequencename** from the current working directory, which becomes the 'currently loaded sequence'. While Siril sequence commands require the sequence name to be specified, it is necessary to load a sequence in order to read its metadata using Python scripting

log

```
log
```

Calcule et applique une échelle logarithmique à l'image chargée, en utilisant la formule suivante : log(1 - (valeur - min) / (max - min)), min et max étant les valeurs minimale et maximale du pixel pour le canal

ls

```
ls
```

Liste des fichiers et répertoires dans le répertoire de travail

Cette commande est seulement disponible sur les systèmes reposants sur Unix, pour une commande équivalente sur Microsoft Windows, voir *dir*.

makepsf

```
makepsf clear
makepsf load filename
makepsf save [filename]
makepsf blind [-l0] [-si] [-multiscale] [-lambda=] [-comp=] [-ks=] [-savepsf=]
makepsf stars [-sym] [-ks=] [-savepsf=]
makepsf manual { -gaussian | -moffat | -disc | -airy } [-fwhm=] [-angle=] [-ratio=] [-
↪beta=] [-dia=] [-fl=] [-wl=] [-pixelsize=] [-obstruct=] [-ks=] [-savepsf=]
```

Génère une PSF à utiliser avec la déconvolution, l'une des trois méthodes exposées par les commandes RL, SB ou WIENER. L'un des éléments suivants doit être fourni comme premier argument : **clear** (efface la PSF existante), **load** (charge une PSF à partir d'un fichier), **save** (sauvegarde la PSF actuelle), **blind** (estimation aveugle de la PSF), **stars** (génère une PSF basée sur les étoiles mesurées dans l'image) ou **manual** (génère une PSF manuellement basée sur une fonction et des paramètres).

Aucun argument additionnel n'est requis quand l'argument **clear** est utilisé.

Pour charger une PSF précédemment sauvegardée, l'argument **load** nécessite le nom du fichier *filename* de la PSF comme second argument. Ce fichier peut être dans n'importe quel format pris en charge par Siril, mais l'image doit être au format carré et idéalement impair.

Pour sauvegarder une PSF précédemment générée, l'argument **save** est utilisé. Un nom de fichier peut être fourni en option (il doit avoir l'une des extensions ".fit", ".fits", ".fts" ou ".tif"), mais si aucun nom n'est fourni, la PSF sera nommée en fonction du nom du fichier ou de la séquence ouvert(e).

Pour l'argument **blind**, les arguments optionnels suivants peuvent être fournis : **-l0** utilise la méthode de descente l0, **-si** utilise la méthode d'irrégularité spectrale, **-multiscale** configure la méthode l0 pour effectuer une estimation PSF multi-échelle, **-lambda=** fournit la constante de régularisation.

Pour les PSF des **étoiles** détectées, le seul paramètre optionnel est **-sym**, qui configure la PSF pour qu'elle soit symétrique.

Pour une PSF **manuelle**, un des arguments **-gaussian**, **-moffat**, **-disc** or **-airy** peut être fourni afin de spécifié la fonction PSF. Pour la PSF Gaussienne ou Moffat les arguments optionels **-fwhm=**, **-angle=** and **-ratio=** peuvent être

ajoutés. Pour la PSF Moffat l'argument optionel **-beta=** peut aussi être ajouté. Si ces valeurs sont omises, elles prennent par défaut les mêmes valeurs que dans fenêtre déconvolution. Pour les PSF "disc" seul l'argument **-fwhm=** est nécessaire, qui, pour cette fonction est utilisé pour donner le *diamètre* de la PSF. Pour les PSF "airy" les arguments suivant peuvent être ajoutés : **-dia=** (définit le diamètre du télescope), **-fl=** (définit la longueur focale du télescope), **-wl=** (définit la longueur d'onde pour calculer la tache d'Airy), **-pixelsize=** (définit la taille des pixels du capteur), **-obstruct=** (définit le l'obstruction centrale comme pourcentage de la surface totale d'ouverture). Si ces paramètres ne sont pas fournis, la longueur d'onde sera par défaut de 525nm et l'obstruction centrale sera par défaut de 0%. Siril tentera de lire les autres paramètres de l'image ouverte, mais certains logiciels d'imagerie peuvent ne pas les fournir tous, auquel cas vous obtiendrez de mauvais résultats, et notez que les métadonnées peuvent ne pas être remplies pour les vidéos au format SER. Vous apprendrez par l'expérience quels sont les éléments sûrs à omettre pour votre configuration d'imagerie particulière.

Pour toutes les options de génération de PSF ci-dessus, l'argument optionnel **-ks=** peut être fourni pour définir la dimension de la PSF, et l'argument optionnel **-savepsf=nom de fichier** peut être utilisé pour sauvegarder la PSF générée : un nom de fichier doit être fourni et les mêmes exigences d'extension de nom de fichier s'appliquent que pour **makepsf save filename**

Liens : *psf*, *rl*, *sb*, *wiener*

merge

```
merge sequence1 sequence2 [sequence3 ...] output_sequence
```

Merges several sequences of the same type (FITS images, FITS sequence or SER) and same image properties into a new sequence with base name **newseq** created in the current working directory, with the same type. The input sequences can be in different directories, can specified either in absolute or relative path, with the exact .seq name or with only the base name with or without the trailing '_'

merge_cfa

```
merge_cfa file_CFA0 file_CFA1 file_CFA2 file_CFA3 bayerpattern
```

Builds a Bayer masked color 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]
```

Retourne l'image chargée autour de l'axe horizontal. L'option **-bottomup** ne retournera l'image que si elle n'est pas bottom-up

mirrorx_single

```
mirrorx_single image
```

Retourne l'image autour de l'axe horizontal, uniquement si nécessaire (si elle n'est pas déjà bottom-up). Siril prend le nom du fichier image comme argument, ce qui lui permet d'éviter de lire les données de l'image si aucune inversion n'est nécessaire. L'image est écrasée si un retournement est effectué

mirrory

```
mirrory
```

Retourne l'image autour de l'axe vertical

modasinh

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

Étirement arcsinh modifié basé sur le travail de l'équipe ghsastro.co.uk.

L'argument **-D=** définit la force de l'étirement, entre 0 et 10. C'est le seul argument obligatoire. Les arguments facultatifs suivants permettent de mieux adapter l'étirement :
LP définit une plage de préservation des ombres entre 0 et SP où l'étirement sera linéaire, préservant les détails d'ombre ;
SP définit le point de symétrie de l'étirement, entre 0 et 1, qui est le point auquel l'étirement est le plus intense ;
HP définit une région entre HP et 1 où l'étirement est linéaire, préservant les détails de la mise en évidence et évitant le gonflement des étoiles.

Si elle est omise, LP et SP ont une valeur par défaut de 0.0 et HP a une valeur par défaut de 1.0.
Un argument facultatif (soit **-human**, **-even** ou **-independent**, peut être passé pour sélectionner la luminance pondérée humaine ou pondérée paire ou les canaux de couleur indépendants pour les étirements de couleur). L'argument est ignoré pour les images mono. Alternativement, l'argument **-sat** spécifie que l'étirement est effectué sur la saturation de l'image - l'image doit être en couleur et tous les canaux doivent être sélectionnés pour que cela fonctionne.
En option, le paramètre **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, V, B, RV, RB ou VB. La valeur par défaut est « tous les canaux ». Le mode clip peut être défini à l'aide de l'argument **-clipmode=** : les valeurs **clip**, **rescale**, **rgbblend** ou **globalrescale** sont acceptées et la valeur par défaut est rgbblend

mtf

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

Applique la fonction de transfert des demi-tons à l'image actuellement chargée.

Trois paramètres sont nécessaires, **low**, **midtones** et **high** où le paramètre de balance des tons moyens définit un étirement non linéaire de l'histogramme dans la plage [0,1]. Pour une détermination automatique des paramètres, voir AUTOSTRETCH.
Le paramètre optionnel **[channels]** peut être utilisé pour spécifier les canaux auxquels appliquer l'étirement : il peut s'agir de R, G, B, RG, RB ou GB. La valeur par défaut est tous les canaux

Liens : *autostretch*

neg

```
neg
```

Modifie les valeurs de pixels de l'image actuellement chargée en une vue négative, comme 1-valeur pour 32 bits, ou 65535-valeur pour 16 bits. Cela ne modifie pas le mode d'affichage

new

```
new width height nb_channel [filename]
```

Crée une nouvelle image remplie de zéros avec une taille de **width** x **height**.

Le paramètre optionnel **[filename]** peut être utilisé pour spécifier le nom du nouveau fichier.

L'image est au format 32 bits et contient **nb_channel** canal(aux) , **nb_channel** étant 1 ou 3. Elle n'est pas sauvegardée, mais devient l'image chargée et elle est affichée et peut être sauvegardée par la suite

nozero

```
nozero level
```

Remplace les valeurs nulles par des valeurs de **level**. Utile avant une opération idiv ou fdiv, principalement pour les images 16 bits

offline

```
offline
```

Met Siril en mode hors ligne. Dans ce mode, les fonctions réseau telles que la consultation de catalogues à distance, la mise à jour de dépôts git, etc. sont indisponibles. Les données mises en cache restent accessibles

offset

```
offset value
```

Ajoute une valeur constante **value** (spécifié en ADU) à l'image actuelle. Cette constante peut prendre une valeur négative.

En mode 16 bits, les valeurs de pixels qui se situent en dehors de l'intervalle [0, 65535] sont rognées (clippées). En mode 32 bits, aucune rognure (clipping) ne se produit

en ligne

```
online
```

Met Siril en mode en ligne. Dans ce mode, les fonctions réseau telles que la consultation de catalogues à distance, la mise à jour de dépôts git, etc. sont autorisées

parse

```
parse str [-r]
```

Analyse la chaîne de caractères **str** en utilisant les informations contenues dans l'en-tête de l'image actuellement chargée. Le but principal de cette commande est de débuguer l'interprétation d'entête des clés d'en-tête qui peuvent être utilisées dans d'autres commandes.
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.

Si la chaîne **str** commence par le préfixe *$def*, elle sera reconnue comme un mot-clé réservé et recherchée dans les chaînes stockées dans gui_prepro.dark_lib, gui_prepro.flat_lib, gui_prepro.bias_lib ou gui_prepro.stack_default pour *$defdark*, *$defflat*, *$defbias* ou *$defstack* respectivement.
Le mot-clé *$seqname$* peut également être utilisé quand une séquence est chargée

pcc

```
pcc [-limitmag=[+-]] [-catalog=] [-bgtol=lower,upper]
```

Exécute la Correction de Couleur par Photométrie sur l'image chargée et résolue astrométriquement.

La magnitude limite des étoiles est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement une valeur positive absolue pour la magnitude limite.
The star catalog used is NOMAD by default, it can be changed by providing **-catalog=apass**, **-catalog=localgaia** or **-catalog=gaia**. If installed locally, the remote NOMAD (the complete version) can be forced by providing **-catalog=nomad**
La tolérance aux valeurs aberrantes de la référence de fond de ciel peut être spécifiée en unités sigma en utilisant **-bgtol=lower,upper** : ces valeurs sont par défaut de -2,8 et +2,0

platesolve

```
platesolve [-force] [image_center_coords] [-focal=] [-pixelsize=]
platesolve ... [-noflip] [-downscale] [-order=] [-radius=] [-disto=]
platesolve ... [-limitmag=[+-]] [-catalog=] [-nocrop]
platesolve ... [-localasnet [-blindpos] [-blindres]]
```

Résolution astrométrique de l'image chargée.

Si l'image a déjà été résolue, rien ne sera fait, à moins que l'argument **-force** ne soit passé pour forcer une nouvelle résolution. Si les mots clés WCS ou d'autres métadonnées de l'image sont erronés ou manquants, des arguments doivent être passés :

les coordonnées approximatives du centre de l'image peuvent être fournies en degrés décimaux ou en degrés/heure minute seconde (J2000 avec des séparateurs de deux points), avec les valeurs d'ascension droite et de déclinaison séparées par une virgule ou un espace (non obligatoire pour astrometry.net).

La longueur focale et la taille des pixels peuvent être passées avec **-focal=** (en mm) et **-pixelsize=** (en microns), en remplaçant les valeurs de l'image et des paramètres. Voir aussi les options pour résoudre en aveugle avec Astrometry.net en local

Sauf si **-noflip** est spécifié, si l'image est détectée comme étant à l'envers, elle sera retournée.

Pour une détection plus rapide des étoiles dans les grandes images, il est possible de sous-échantillonner l'image avec **-downscale**.

La résolution peut prendre en compte les distorsions en utilisant la convention SIP avec des polynômes jusqu'à l'ordre 5. La valeur par défaut est prise dans les préférences astrométriques. Elle peut être modifiée avec l'option **-order=** qui donne une valeur entre 1 et 5.

Lors de l'utilisation des catalogues locaux du solveur Siril ou avec Astrometry.net local, si la résolution initiale n'est pas réussie, le solveur cherchera une solution dans un cône de rayon spécifié avec l'option **-radius=**. Si aucune valeur n'est fournie, le rayon de recherche est pris dans les préférences d'astrométrie. La recherche Siril par proximité peut être désactivée en passant une valeur de 0 (ne peut pas être désactivée pour Astrometry.net).

Vous pouvez sauvegarder la solution courante comme fichier de distorsion avec l'option **-disto=**.

Les images peuvent être résolues par Siril à l'aide d'un catalogue d'étoiles et de l'algorithme d'alignement global ou par la commande locale solve-field d'astrometry.net (activée avec **-localasnet**).

Options du solveur astrométrique de Siril :

La magnitude limite des étoiles utilisées pour la résolution astrométrique et le PCC est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement avec une valeur absolue positive pour la magnitude limite.

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 catalog given in argument, with possible values : tycho2, nomad, localgaia, gaia, ppmxl, brightstars, apass.

Si le champ de vision calculé est supérieur à 5 degrés, la détection des étoiles sera limitée à une zone recadrée autour du centre de l'image, à moins que l'option **-nocrop** ne soit passée.

Options du solveur Astrometry.net :

Les options **-blindpos** et/ou **-blindres** permettent de résoudre en aveugle la position et la résolution respectivement. Vous pouvez utiliser ces options pour résoudre une image dont la position et l'échantillonnage sont totalement inconnus

pm

```
pm "expression" [-rescale [low] [high]] [-nosum]
```

Cette commande évalue l'expression donnée en argument comme dans l'outil PixelMath. L'expression complète doit être entre guillemets et les variables (qui sont des noms d'images, sans extension, situés dans le répertoire de travail dans ce cas) doivent être entourées du symbole $, par exemple "$image1$ * 0.5 + $image2$ * 0.5". Un maximum de 10 images peut être utilisé dans l'expression.
L'image peut être redimensionnée avec l'option **-rescale** suivie des valeurs **low** et **high** dans l'intervalle [0, 1]. Si aucune valeur basse ou haute n'est fournie, les valeurs par défaut sont 0 et 1. Un autre argument optionnel, **-nosum**, indique à Siril de ne pas additionner les temps d'exposition. Cela a un impact sur les mots-clés FITS tels que LIVETIME et STACKCNT

profile

```
profile -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] { [-savedat] | [-filename=] } [-
↪layer=] [-width=] [-spacing=] ["-title=My Plot"]
```

Génère un tracé du profil d'intensité entre 2 points de l'image, également connu sous le nom de *coupure*. Les arguments peuvent être fournis dans n'importe quel ordre. Les arguments **-to=x,y** et **-from=x,y** sont obligatoires.

The argument **-layer={red | green | blue | lum | col}** specifies which channel (or luminance or colour) to plot if the image is color. It may also be used with the **-tri** option, which generates 3 parallel equispaced profiles each separated by **-spacing=** pixels, but note that for tri profiles the **col** option will be treated the same as **lum**.

L'option **-cfa** sélectionne le mode CFA, qui génère 4 profils : 1 pour chaque canal CFA dans une image avec motif de Bayer. Cette option ne peut pas être utilisée avec des images couleur ou des images mono sans motif de Bayer, et ne peut pas être utilisée en même temps que l'option **-tri**.

The option **-arcsec** causes the x axis to display distance in arcsec, if the necessary metadata is available. If not provided or if metadata is not available, distance will be shown in pixel units.

L'argument **-savedat** permet de sauvegarder les fichiers de données : le nom du fichier sera écrit dans le log.
L'argument **-filename=** peut également être utilisé pour spécifier un nom de fichier dans lequel écrire le fichier de données. (L'option **-filename=** implique **-savedat**.)

L'argument **"-title=Mon Titre "** définit un titre personnalisé "Mon Titre"

psf

```
psf [channel]
```

Effectue une PSF (Fonction d'étalement du point) sur l'étoile sélectionnée et affiche les résultats. Pour les opérations en mode console, la sélection peut être donnée en pixels en utilisant BOXSELECT. S'il est fourni, l'argument **canal** sélectionne le canal de l'image sur lequel l'étoile sera analysée. Il peut être omis pour les images monochromes ou lorsqu'il est exécuté à partir de l'interface graphique avec l'un des canaux actifs dans la vue

Liens : *boxselect*

pwd

```
pwd
```

Imprime le répertoire de travail actuel

pyscript

```
pyscript scriptname.py [script_argv]
```

Exécute un script Siril en python

Le nom du script doit être fourni en tant que premier argument. S'il n'est pas trouvé dans le répertoire de travail actuel, les chemins d'accès aux scripts définis par l'utilisateur spécifiés dans les préférences et le répertoire local siril-scripts seront recherchés. Tous les arguments suivants seront traités comme des arguments de script et transmis au script sous forme de vecteur d'arguments. Notez que le script spécifique doit prendre en charge la lecture des entrées à partir du vecteur d'arguments

register

```
register sequencename [-2pass] [-selected] [-prefix=] [-scale=]
register sequencename ... [-layer=] [-transf=] [-minpairs=] [-maxstars=] [-nostarlist] [-
↪disto=]
register sequencename ... [-interp=] [-noclamp]
register sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
```

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.

Toutes les images de la séquence seront alignées sauf si l'option **-selected** est passée, dans ce cas les images exclues ne seront pas traitées.
The **-2pass** option 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, based on image quality and framing. To generate transformed images after this pass, use SEQAPPLYREG.
If created, the output sequence name will start with the prefix "r_" unless otherwise specified with **-prefix=** option. The output images can be rescaled by passing a **-scale=** argument with a float value between 0.1 and 3.

Options de transformation d'image :

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.
-transf= specifies the use of either **shift**, **similarity**, **affine** or **homography** (default) transformations respectively.
-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.
-maxstars= will specify the maximum number of stars 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.
-nostarlist désactive la sauvegarde de la liste d'étoiles sur le disque.
-disto= uses distortion terms from a previous platesolve solution (with a SIP order > 1). It takes as parameter either **image** to use the solution contained in the currently loaded image, **file** followed by the path to the image containing the solution or **master** to load automatically the matching distortion master corresponding to each image. When using this option, the polynomials are used both to correct star positions before computing the transformation and to undistort the images when output images are exported.

Option d'interpolation de l'image :

Par défaut, les transformations sont appliquées pour aligner les images à l'aide de l'interpolation.
La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.
La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**.

Options de drizzle pour les images :

Sinon, les images peuvent être exportées à l'aide de l'algorithme HST drizzle en passant l'argument **-drizzle** qui peut prendre les options supplémentaires suivantes :
-pixfrac= défini la fraction de pixel (par défaut =1.0).
L'argument **-kernel=** défini le noyau de drizzle et doit être suivi par l'un des arguments suivant : **point**, **turbo**, **square**, **gaussian**, **lanczos2** or **lanczos3**. La valeur par défaut est **square**.
L'argument **-flat=** spécifie un flat maître pour pondérer les pixels d'entrée drizzlés (la valeur par défaut est sans flat).

Remarque : lorsque vous utilisez **-drizzle** sur des images prises avec un capteur couleur, les images d'entrée ne doivent pas être débayerisées. Dans ce cas, la détection des étoiles se fera toujours sur les pixels verts

Liens : *setfindstar*, *psf* , *seqapplyreg*

reloadscripts

```
reloadscripts
```

Réanalyse les dossiers de scripts et met à jour le menu Scripts. Notez que cette commande ne peut pas être utilisée en mode ligne de commande (c'est-à-dire à partir de siril-cli) et que la mise à jour s'effectuera de manière asynchrone (c'est-à-dire que la commande déclenchera la mise à jour, mais n'attendra pas qu'elle soit terminée)

nécessite

```
requires min_version [obsolete_version]
```

Renvoie une erreur si la version de Siril est antérieure à la version minimale requise passée dans le premier argument. En option, prend un deuxième argument pour la version de Siril à partir de laquelle le script est obsolète : renvoie une erreur si la version de Siril est **supérieure ou égale** à celle passée dans le deuxième argument.

Exemple : *requires 1.2.0 1.4.0* permet au script de s'exécuter pour toutes les séries 1.2.x et 1.3.x, mais ne s'exécutera pas pour les versions antérieures à 1.2.0, ni pour la version 1.4.0 ou toute version ultérieure

resample

```
resample { factor | -width= | -height= | -maxdim= } [-interp=] [-noclamp]
```

Resamples the loaded image, either with a factor **factor** or for the target width or height provided by either of **-width=**, **-height=** or **-maxdim=**. This is generally used to resize images : a factor of 0.5 divides size by 2. The **-maxdim** argument can be used to resize the longest dimension of the image to a set size, which can be useful for optimizing images for certain websites, e.g. social media websites.
Dans l'interface graphique utilisateur, nous pouvons voir que plusieurs algorithmes d'interpolation sont proposés.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi d'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}.
La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**

rgbcomp

```
rgbcomp red green blue [-out=result_filename] [-nosum]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename] [-nosum]
```

Crée une composition RVB à l'aide de trois images indépendantes, ou une composition LRVB à l'aide de l'image de luminance optionnelle et de trois images monochromes ou d'une image couleur. L'image résultante est appelée composed_rgb.fit ou composed_lrgb.fit à moins qu'un autre nom ne soit fourni dans l'argument optionnel. Un autre argument optionnel, **-nosum**, indique à Siril de ne pas additionner les temps d'exposition. Cela a un impact sur les mots-clés FITS tels que LIVETIME et STACKCNT

rgradient

```
rgradient xc yc dR dalpha
```

Crée deux images, avec un décalage radial (**dR** en pixels) et un décalage rotatif (**dalpha** en degrés) par rapport au point (**xc**, **yc**).

Entre ces deux images, le décalage a la même amplitude mais est de signe opposé. Ces deux images sont additionnées pour créer l'image finale. Ce processus est aussi appelé filtre Larson Sekanina

rl

```
rl [-loadpsf=] [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Restaure une image en utilisant la méthode Richardson-Lucy.

En option, une PSF peut être chargé en utilisant l'argument **-loadpsf=filename** (créé avec MAKEPSF).

Le nombre d'itérations est donné par **-iters** (par défaut 10).

Le type de régularisation peut être défini avec **-tv** pour la variation totale, ou **-fh** pour la norme de Frobenius de la matrice hessienne (la valeur par défaut est aucune) et **-alpha=** fournit la force de régularisation (une valeur plus faible = plus de régularisation, par défaut = 3000).

Par défaut, la méthode du gradient descendant est utilisé avec une taille de pas par défaut de 0.0005, cependant la méthode multiplicative peut être spécifiée avec **-mul**.

Le critère d'arrêt peut être activé en spécifiant une limite d'arrêt avec **-stop=**

Liens : *psf* , *makepsf*

rmgreen

```
rmgreen [-nopreserve] [type] [amount]
```

Applique un filtre de réduction de bruit chromatique. Il supprime la teinte verte de l'image actuelle. Ce filtre est basé sur SCNR de PixInsight et c'est aussi le même filtre utilisé par le plugin HLVG de Photoshop.
La luminosité est préservée par défaut mais peut être désactivé avec le paramètre **-nopreserve**.

Type peut prendre les valeurs 0 pour neutre moyen, 1 pour neutre maximum, 2 pour masque maximum, 3 pour masque additif, par défaut à 0. Les deux derniers peuvent prendre un argument **amount**, une valeur entre 0 et 1, la valeur par défaut étant 1

rotate

```
rotate degree [-nocrop] [-interp=] [-noclamp]
```

Fait pivoter l'image chargée d'un angle de **degree** degrés. L'option **-nocrop** peut être ajoutée pour éviter le recadrage à la taille de l'image (des bordures noires seront ajoutées).

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.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.
La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**

rotatePi

```
rotatePi
```

Fait pivoter l'image chargée d'un angle de 180° autour de son centre. Ceci est équivalent à la commande "ROTATE 180" ou "ROTATE -180"

Liens : *rotate*

satu

```
satu amount [background_factor [hue_range_index]]
```

Renforce la saturation des couleurs de l'image chargée. Essayez de manière itérative pour obtenir les meilleurs résultats.

amount peut être un nombre positif pour augmenter la saturation des couleurs, négatif pour la réduire, 0 ne fait rien, 1 l'augmente de 100%

background_factor est un facteur de (médiane + écart-type) utilisé pour définir un seuil, de sorte que seuls les pixels au-dessus de ce seuil seraient modifiés. Cela permet d'avoir un bruit de fond non saturé en couleur, si le choix est fait avec soin. La valeur par défaut est de 1. Une valeur de 0 désactive le seuil.

hue_range_index peut être entre [0, 6], ce qui signifie : 0 pour rose à orange, 1 pour orange à jaune, 2 pour jaune à cyan, 3 pour cyan, 4 pour cyan à magenta, 5 pour magenta à rose, 6 pour tout (par défaut)

save

```
save filename [-chksum]
```

Saves current image to **filename**.fit (or .fits, depending on your preferences, see SETEXT) in the current working directory. The image remains loaded. **filename** can contain a path as long as the directory already exists. The **-chksum** option stores checksum keywords (CHECKSUM and DATASUM) in the FITS header

Liens : *setext*

savebmp

```
savebmp filename
```

Sauvegarde l'image courante sous forme d'un fichier bitmap avec 8 bit par canal : **filename**.bmp (BMP 24-bit)

savejpg

```
savejpg filename [quality]
```

Sauvegarde l'image courante dans un fichier JPG : **filename**.jpg.

La qualité de la compression peut être ajustée à l'aide de la valeur optionnelle **quality**, 100 étant la meilleure valeur par défaut, tandis qu'une valeur inférieure augmente le taux de compression

savejxl

```
savejxl filename [-effort=] [-quality=] [-8bit]
```

Sauvegarde l'image courante dans un fichier JPG XL : **filename**.jxl.

Tous les autres arguments sont facultatifs. Le paramètre de qualité exprime une distance maximale admissible entre l'image originale et l'image compressée : l'argument **-quality=** peut être fourni et doit être spécifié sous la forme d'un nombre à virgule flottante compris entre 0,0 et 10,0. Une qualité plus élevée signifie une meilleure qualité, mais une taille de fichier plus importante. Qualité = 10.0 est mathématiquement sans perte, qualité = 9.0 est visuellement sans perte et qualité = 0 est visuellement médiocre, mais donne des fichiers de très petite taille. La valeur par défaut est 9,0 ; les valeurs typiques sont comprises entre 7,0 et 10,0. L'effort de compression peut être ajusté en utilisant la valeur optionnelle **-effort=**, 9 étant l'effort le plus important, mais très lent, tandis qu'une valeur plus faible augmente le taux de compression. Les valeurs supérieures à 7 ne sont pas recommandées, car elles peuvent être très lentes et n'apporter que peu ou pas d'avantages à la taille du fichier. En fait, un effort = 9 peut parfois produire des fichiers plus volumineux. Si cet argument est omis, la valeur par défaut de 7 est utilisée. Une option **-8bit** peut être fournie pour forcer la sortie à 8 bits par pixel

savepng

```
savepng filename
```

Sauvegarde l'image courante dans un fichier PNG : **filename**.png, avec 16 bits par canal si l'image est chargée en 16 ou 32 bits, et en 8 bits par canal si l'image est chargée en 8 bits

savepnm

```
savepnm filename
```

Enregistre l'image actuelle sous la forme d'un format de fichier NetPBM avec 16 bits par canal.

L'extension de sortie doit être **filename**.ppm pour une image RVB et **filename**.pgm pour une image en nuance de gris

savetif

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

Enregistre l'image actuelle sous la forme d'un fichier TIFF non compressé de 16 bits par canal : **filename**.tif. L'option **-astro** permet d'enregistrer au format Astro-TIFF, tandis que **-deflate** active la compression.

Voir aussi SAVETIF32 et SAVETIF8

savetif32

```
savetif32 filename [-astro] [-deflate]
```

Même commande que SAVETIF mais le fichier de sortie est enregistré en 32 bits par canal : **filename**.tif. L'option **-astro** permet de sauvegarder au format Astro-TIFF, tandis que **-deflate** permet la compression

Liens : *savetif*

savetif8

```
savetif8 filename [-astro] [-deflate]
```

Même commande que SAVETIF mais le fichier de sortie est enregistré en 8 bits par canal : **filename**.tif. L'option **-astro** permet de sauvegarder au format Astro-TIFF, tandis que **-deflate** permet la compression

Liens : *savetif*

sb

```
sb [-loadpsf=] [-alpha=] [-iters=]
```

Restaure une image en utilisant la méthode Split Bregman.

En option, une PSF peut être chargée en utilisant l'argument **-loadpsf=nom de fichier**.

Le nombre d'itération est donné par **-iters** (par défaut 1).

Le facteur de régularisation **-alpha=** fournit la force de la régularisation (plus petite valeur = plus de régularisation, par défaut = 3000)

Liens : *psf*

select

```
select sequencename from to
```

Cette commande permet de sélectionner facilement en masse des images dans la séquence **sequencename** (de **from** à **to** inclus). Il s'agit d'une sélection en vue d'un traitement ultérieur.
Voir aussi UNSELECT

Liens : *unselect*

Exemples :

select . 0 0
sélectionne le premier de la séquence actuellement chargée

select sequencename 1000 1200
sélectionne 201 images à partir du numéro 1000 dans la séquence nommée sequencename

Le deuxième nombre peut être supérieur au nombre d'images pour aller jusqu'au bout.

seqapplyreg

```
seqapplyreg sequencename [-prefix=] [-scale=] [-layer=] [-framing=]
seqapplyreg sequencename ... [-interp=] [-noclamp]
seqapplyreg sequencename ... [-drizzle [-pixfrac=] [-kernel=] [-flat=]]
seqapplyreg sequencename ... [-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-
↪filter-round=value[%|k]] [-filter-bkg=value[%|k]] [-filter-nbstars=value[%|k]] [-
↪filter-quality=value[%|k]] [-filter-incl[uded]]
```

Applique des transformations géométriques aux images de la séquence donnée en argument afin qu'elles puissent être superposées à l'image de référence, en utilisant les données d'alignement précédemment calculées (voir REGISTER).

Le nom de la séquence de sortie commence par le préfixe **"r_"** sauf indication contraire spécifié par l'option **-prefix=**.

L'alignement est réalisé sur la première couche pour laquelle les données existe pour une image RVB sauf si spécifié par l'option **-layer=** (0, 1 ou 2 pour R, V et B respectivement).

The output images can be rescaled by passing a **-scale=** argument with a float value between 0.1 and 3.

Le cadrage automatique de la séquence de sortie peut être spécifié en utilisant le mot-clé **-framing=** suivi de l'une des méthodes de la liste { current | min | max | cog } :

-framing=max (bounding box) will project each image and compute its shift wrt. reference image. The resulting sequence can then be stacked using option **-maximize** of STACK command which will create the full image encompassing all images of the sequence.

-framing=min (zone commune) recadre chaque image dans la zone qu'elle a en commun avec toutes les images de la séquence.

-framing=cog détermine la meilleur position de cadrage comme centre de gravité (cog) de toutes les images.

Option d'interpolation de l'image :

Par défaut, les transformations sont appliquées pour aligner les images à l'aide de l'interpolation.

La méthode d'interpolation des pixels peut être spécifiée avec l'argument **-interp=** suivi de l'une des méthodes de la liste **no**[ne], **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}. Si **none** est passé, la transformation est forcée en translation et un décalage par pixel est appliqué à chaque image sans aucune interpolation.

La contrainte des méthodes d'interpolation bicubique et lanczos4 est la valeur par défaut, pour éviter les artefacts, mais peut être désactivé avec l'argument **-noclamp**.

Options de drizzle pour les images :

Sinon, les images peuvent être exportées à l'aide de l'algorithme HST drizzle en passant l'argument **-drizzle** qui peut prendre les options supplémentaires suivantes :

-pixfrac= défini la fraction de pixel (par défaut =1.0).

L'argument **-kernel=** défini le noyau de drizzle et doit être suivi par l'un des arguments suivant : **point**, **turbo**, **square**, **gaussian**, **lanczos2** or **lanczos3**. La valeur par défaut est **square**.

L'argument **-flat=** spécifie un flat maître pour pondérer les pixels d'entrée drizzlés (la valeur par défaut est sans flat).

Filtrage des images :

Les images qui vont être aligner peuvent être sélectionnées en utilisant des filtres, tel que celles sélectionné ou avec la meilleur FWHM, avec l'une des option **-filter-***.

Links : *register*, *stack*

Avec le filtrage étant l'un de ceux-ci, sans ordre ni nombre particulier :

```
[-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]]
```

Les meilleurs images de la séquence peuvent être empilées en utilisant les arguments de filtrage. Chacun de ces argument peu supprimer les mauvaises images sur la base des propriétés que leur nom contient, récupéré depuis les données d'alignement, avec l'un des trois types de valeur d'argument :
- 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.

Il est également possible d'utiliser des images sélectionnées manuellement, soit précédemment à partir de l'interface graphique utilisateur (GUI), soit avec les commandes select ou unselect, en utilisant l'argument **-filter-included**.

seqccm

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

Même commande que CCM mais pour la séquence **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

Le nom de la séquence de sortie commence par le préfixe "ccm_", sauf indication contraire avec l'option **-prefix=**

Liens :*ccm*

seqclean

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

Cette commande efface les données de sélection, d'alignement et/ou de statistiques stockées pour la séquence **sequencename**.

Vous pouvez préciser de n'effacer que l'alignement, les statistiques et/ou la sélection avec les options **-reg**, **-stat** et **-sel**, respectivement. Tout est effacé si aucune option n'est passée

seqcosme

```
seqcosme sequencename [filename].lst [-prefix=]
```

Même commande que COSME mais pour la séquence **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

Le nom de la séquence de sortie commence par le préfixe "cosme_", sauf indication contraire avec l'option **-prefix=**

Liens :*cosme*

seqcosme_cfa

```
seqcosme_cfa sequencename [filename].lst [-prefix=]
```

Même commande que COSME_CFA mais pour la séquence **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

Le nom de la séquence de sortie commence par le préfixe "cosme_", sauf indication contraire avec l'option **-prefix=**

Liens : *cosme_cfa*

seqcrop

```
seqcrop sequencename x y width height [-prefix=]
```

Recoupe la séquence donnée en argument **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

La sélection est spécifiée par la position du coin supérieur gauche **x** et **y** et la sélection **width** et **height**, comme pour CROP.
Le nom de la séquence de sortie commence par le préfixe "cropped_" sauf indication contraire avec l'option **-prefix=**

Liens : *crop*

seqextract_Green

```
seqextract_Green sequencename [-prefix=]
```

Même commande que EXTRACT_GREEN mais pour une séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "Green_" sauf indication contraire avec l'option **-prefix=**

seqextract_Ha

```
seqextract_Ha sequencename [-prefix=] [-upscale]
```

Même commande que EXTRACT_HA mais pour une séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "Ha_" sauf indication contraire avec l'option **-prefix=**

seqextract_HaOIII

```
seqextract_HaOIII sequencename [-resample=]
```

Même commande que EXTRACT_HAOIII mais pour une séquence **sequencename**.

Le nom de la séquence de sortie comme par le préfixe "Ha_" et "OIII_"

seqfind_cosme

```
seqfind_cosme sequencename cold_sigma hot_sigma [-prefix=]
```

Même commande que FIND_COSME mais pour une séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "cc_" sauf indication contraire avec l'option **-prefix=**

Liens : *find_cosme*

seqfind_cosme_cfa

```
seqfind_cosme_cfa sequencename cold_sigma hot_sigma [-prefix=]
```

Même commande que FIND_COSME_CFA mais pour une séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "cc_" sauf indication contraire avec l'option **-prefix=**

Liens : *find_cosme_cfa*

seqfindstar

```
seqfindstar sequencename [-layer=] [-maxstars=]
```

Même commande que FINDSTAR mais pour une séquence **sequencename**.

L'option **-out=** n'est pas disponible pour ce processus car tous les fichiers de liste d'étoiles sont sauvegardés avec le nom par défaut *seqname_seqnb.lst*

Liens : *findstar*

seqfixbanding

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

Même commande que FIXBANDING mais pour la séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "unband_", sauf indication contraire avec l'option **-prefix=**

Liens : *fixbanding*

seqght

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

Même commande que GHT, mais la séquence doit être spécifiée comme premier argument. En outre, l'argument optionnel **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *ght*

seqheader

```
seqheader sequencename keyword [keyword2 ...] [-sel] [-out=file.csv]
```

Prints the FITS header value corresponding to the given keys for all images in the sequence. You can write several keys in a row, separated by a space. The **-out=** option, followed by a file name, allows you to print the output in a csv file. The **-sel** option limits the output to the images selected in the sequence

seqinvght

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

Identique à la commande INVGHT, mais s'applique à une séquence qui doit être spécifiée comme premier argument. En outre, l'argument facultatif **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *invght*

seqinvmodasinh

```
seqinvmodasinh sequence -D= [-LP=] [-SP=] [-HP=] [-clipmode=] [-human | -even | -
↪independent | -sat] [channels] [-prefix=]
```

Identique à la commande INVMODASINH, mais s'applique à une séquence qui doit être spécifiée comme premier argument. En outre, l'argument facultatif **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *invmodasinh*

seqlinstretch

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

Identique à la commande INVGHT, mais s'applique à une séquence qui doit être spécifiée comme premier argument. En outre, l'argument facultatif **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *linstretch*

seqmerge_cfa

```
seqmerge_cfa sequencename0 sequencename1 sequencename2 sequencename3 bayerpattern [-
↪prefixout=]
```

Fusionne 4 séquence d'image pour reconstruire la matrice de Bayer. Les séquence sont spécifié dans les arguments **sequencename0**, **sequencename1**, **sequencename2** et **sequencename3**.

The Bayer pattern to be reconstructed must be provided as the second argument as one of RGGB, BGGR, GBRG or GRBG (the order of the Bayer channels must match the order of the specified sequences).

Note : les 4 séquences **doivent** être présente et avoir les mêmes dimension, profondeur de bit et nombre d'images.

Le nom de la séquence de sortie commence par le préfixe "mCFA_" sauf indication contraire avec l'option **-prefixout=**

seqmodasinh

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

Même commande que MODASINH, mais la séquence doit être spécifiée comme premier argument. En outre, l'argument optionnel **-prefix=** peut être utilisé pour définir un préfixe personnalisé

Liens : *modasinh*

seqmtf

```
seqmtf sequencename low mid high [channels] [-prefix=]
```

Même commande que pour MTF mais pour la séquence **sequencename**.

Le nom de la séquence de sortie commence par le préfixe "mtf_", sauf indication contraire avec l'option **-prefix=**

Liens : *mtf*

seqprofile

```
seqprofile sequence -from=x,y -to=x,y [-tri] [-cfa] [-arcsec] [-savedat] [-layer=] [-
↪width=] [-spacing=] [ {-xaxis=wavelength | -xaxis=wavenumber } ] [{-wavenumber1= | -
↪wavelength1=} -wn1at=x,y {-wavenumber2= | -wavelength2=} -wn2at=x,y] ["-title=My Plot"]
```

Génère un profil d'intensité entre 2 points de chaque image de la séquence. Après le premier argument obligatoire indiquant la séquence à traiter, les autres arguments sont les mêmes que pour la commande **profile**. Si l'on traite une séquence et que l'on souhaite que le numéro de l'image courante et le nombre total d'images soient affichés dans le format "Ma séquence (1 / 5)", le titre donné doit se terminer par () (par exemple, "Ma séquence ()" et les nombres seront remplis automatiquement.)

seqpsf

```
seqpsf sequencename [channel] [{ -at=x,y | -wcs=ra,dec }] [-followstar]
```

Même commande que PSF mais fonctionne sur des séquences. Cette commande est similaire à l'alignement sur une étoile, sauf que les résultats peuvent être utilisés pour l'analyse de la photométrie plutôt que pour l'alignement des images et que les coordonnées de l'étoile peuvent être fournies par des options.
This command is what is called internally by the menu that appears on right click in the image, with the PSF for the sequence entry. If registration data already exist for the sequence, they will can be used to shift the search window in each image. If there is no registration data and if there is significant shift between images in the sequence, the default settings will fail to find stars in the initial position of the search area.

L'option de suivi d'étoile peut être activé avec l'argument **-followstar**.

Les résultats sont affichés dans l'onglet Graphique, à partir duquel ils peuvent également être exportés vers un fichier de valeurs séparées par des virgules (CSV) en vue d'une analyse externe.

Lors de la création d'une courbe de lumière, la première étoile pour laquelle seqpsf a été exécuté, marquée "V" à l'écran, sera considérée comme l'étoile variable. Toutes les autres étoiles sont moyennées pour créer une courbe de lumière de référence soustraite à la courbe de lumière de l'étoile variable.

Currently, in headless operation, the command prints some analysed data in the console, another command allows several stars to be analysed and plotted as a light curve : LIGHT_CURVE. Arguments are mandatory : the sequence name must be provided ("." may be used to indicate the currently loaded sequence) and when headless it is mandatory to provide the coordinates of the star, with -at= allowing coordinates in pixels to be provided for the target star of -wcs= allowing J2000 equatorial coordinates to be provided

Liens : *psf* , *light_curve*

seqplatesolve

```
seqplatesolve sequencename [image_center_coords] [-focal=] [-pixelsize=]
seqplatesolve sequencename ... [-downscale] [-order=] [-radius=] [-force] [-noreg] [-
↪disto=]
seqplatesolve sequencename ... [-limitmag=[+-]] [-catalog=] [-nocrop] [-nocache]
seqplatesolve sequencename ... [-localasnet [-blindpos] [-blindres]]
```

Résout en astrométrie une séquence. Une nouvelle séquence sera créée avec le préfixe "ps_" si la séquence d'entrée est SER, sinon, les en-têtes des images seront mis à jour. Dans le cas de SER, il est obligatoire de fournir les métadonnées et la séquence de sortie sera au format FITS cube, car SER ne peut pas stocker les données WCS.

Si les infos WCS ou d'autres métadonnées d'image sont erronés ou manquants, des arguments doivent être transmis :

les coordonnées approximatives du centre de l'image peuvent être fournies en degrés décimaux ou en degrés/heure minute seconde (J2000 avec des séparateurs de deux points), avec les valeurs d'ascension droite et de déclinaison séparées par une virgule ou un espace (non obligatoire pour astrometry.net).

La longueur focale et la taille des pixels peuvent être passées avec **-focal=** (en mm) et **-pixelsize=** (en microns), en remplaçant les valeurs des images et des paramètres. Voir aussi les options pour résoudre en aveugle avec Astrometry.net en local

Pour une détection plus rapide des étoiles dans les grandes images, il est possible de sous-échantillonner l'image avec **-downscale**.

La résolution peut prendre en compte les distorsions en utilisant la convention SIP avec des polynômes jusqu'à l'ordre 5. La valeur par défaut est prise dans les préférences astrométriques. Elle peut être modifiée avec l'option **-order=** qui donne une valeur entre 1 et 5.

Lors de l'utilisation des catalogues locaux du solveur Siril ou avec Astrometry.net local, si la résolution initiale n'est pas réussie, le solveur cherchera une solution dans un cône de rayon spécifié avec l'option **-radius=**. Si aucune valeur n'est fournie, le rayon de recherche est pris dans les préférences d'astrométrie. La recherche Siril par proximité peut être désactivée en passant une valeur de 0 (ne peut pas être désactivée pour Astrometry.net).

Les images déjà résolues seront ignorées par défaut. Ceci peut être désactivé en passant l'option **-force**.

L'utilisation de cette commande mettra à jour les données d'alignement, sauf si l'option **-noreg** est spécifiée.
Vous pouvez sauvegarder la solution courante comme fichier de distorsion avec l'option **-disto=**.

La résolution astrométrique des images peut être faite soit par Siril en utilisant un catalogue d'étoiles et l'algorithme d'alignement global, soit par astrometry.net avec la commande locale solve-field (activée avec **-localasnet**).

Options du solveur astrométrique de Siril :
La magnitude limite des étoiles utilisées pour la résolution astrométrique et le PCC est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement avec une valeur absolue positive pour la magnitude limite.
Le choix du catalogue d'étoiles est automatique sauf si l'option **-catalog=** est passée : si des catalogues locaux sont installés, ils sont utilisés, sinon le choix est basé sur le champ de vision et la magnitude limite. Si l'option est passée, elle force l'utilisation du catalogue distant donné en argument, avec les valeurs possibles : tycho2, nomad, gaia, ppmxl, brightstars, apass.
Si le champ de vision calculé est supérieur à 5 degrés, la détection des étoiles sera limitée à une zone recadrée autour du centre de l'image, à moins que l'option **-nocrop** ne soit passée.
Lors de l'utilisation de catalogues en ligne, une seule extraction de catalogue sera effectuée pour l'ensemble de la séquence. S'il y a beaucoup de dérive ou un échantillonnage différent, cela peut ne pas réussir pour toutes les images. Ceci peut être désactivé en passant l'argument **-nocache**, auquel cas les métadonnées de chaque image seront utilisées (à l'exception des valeurs forcées comme les coordonnées du centre, la taille du pixel et/ou la longueur focale).

Options du solveur Astrometry.net :
Les options **-blindpos** et/ou **-blindres** permettent de résoudre en aveugle la position et la résolution respectivement. Vous pouvez utiliser ces options pour résoudre une image dont la position et l'échantillonnage sont totalement inconnus

seqresample

```
seqresample sequencename { -scale= | -width= | -height= } [-interp=] [-prefix=]
```

Met à l'échelle de la séquence donnée en argument **sequencename**. Seules les images sélectionnées dans la séquence sont traitées.

The scale factor is specified either by the **-scale=** argument or by setting the output width, height or maximum dimension using the **-width=**, **-height=** or **-maxdim=** options.

Une méthode d'interpolation peut être spécifiée en utilisant l'argument **-interp=** suivi d'une des méthodes de la liste **ne**[arest], **cu**[bic], **la**[nczos4], **li**[near], **ar**[ea]}... Le clamping est appliqué pour l'interpolation cubique et lanczos.

Le nom de la séquence de sortie commence par le préfixe "scaled_", sauf indication contraire avec l'option **-prefix=**

seqrl

```
seqrl sequencename [-loadpsf=] [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Identique à la commande RL, mais s'applique à une séquence qui doit être spécifiée comme premier argument

Liens : *rl*

seqsb

```
seqsb sequencename [-loadpsf=] [-alpha=] [-iters=]
```

Identique à la commande SB, mais s'applique à une séquence qui doit être spécifiée comme premier argument

Liens : *sb*

seqsetmag

```
seqsetmag magnitude
```

Identique à la commande SETMAG mais pour la séquence chargée.

Cette commande n'est valable qu'après avoir exécuté SEQPSF ou sa contrepartie graphique (sélectionner la zone autour d'une étoile et lancer l'analyse PSF de la séquence, elle apparaîtra dans les graphiques).
Cette commande a le même but que SETMAG mais recalcule la magnitude de référence pour chaque image de la séquence où l'étoile de référence a été trouvée.
Lors de l'exécution de la commande, la dernière étoile qui a été analysée sera considérée comme l'étoile de référence. L'affichage du graphe de magnitude avant de taper la commande facilite sa compréhension.
Pour réinitialiser l'étoile de référence et le décalage de magnitude, voir SEQUNSETMAG

Liens : *setmag*, *seqpsf* , *psf* , *sequnsetmag*

seqsplit_cfa

```
seqsplit_cfa sequencename [-prefix=]
```

Même commande que SPLIT_CFA mais pour une séquence **sequencename**.

Les noms des séquences de sortie commencent par le préfixe "CFA_" et un nombre, sauf indication contraire avec l'option **-prefix=**.
Limitation : la séquence produit toujours une séquence de fichiers FITS, quel que soit le type de séquence d'entrée

Liens : *split_cfa*

seqstarnet

```
seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Cette commande appelle StarNet++ pour supprimer les étoiles de la séquence **sequencename**. Voir STARNET

Liens : *starnet*

seqstat

```
seqstat sequencename output_file [option] [-cfa]
```

Même commande que STAT pour la séquence **sequencename**.

Les données sont enregistrées dans un fichier csv **fichier_de_sortie**.
Le paramètre optionnel défini le nombre de valeurs statistique calculées : **basic**, **main** (par défaut) ou **full** (plus détaillé mais plus long à calculer).
t\ **basic** inclut mean, median, sigma, bgnoise, min et max
t\ **main** inclut basic avec l'ajout de avgDev, MAD et la racine carrée de BWMV
t\ **full** inclut main avec l'ajout de la position et de l'échelle.

Si **-cfa** est donnée et que l'image est CFA, les statistiques sont réalisées sur les extraction par filtre

Liens : *stat*

seqsubsky

```
seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-
↪smooth=0.5] [-prefix=]
```

Même commande que SUBSKY mais pour la séquence **sequencename**.
Dithering, required for low dynamic gradients, can be disabled with **-nodither**. Note that the **-existing** option is not available for sequence background removal, as the frames of a sequence are not necessarily always aligned.

Le nom de la séquence de sortie commence par le préfixe "bkg_" sauf indication contraire avec l'option **-prefix=**.
Seules les images sélectionnées dans la séquence sont traitées

Liens : *subsky*

seqtilt

```
seqtilt sequencename
```

Même commande que TILT mais pour la séquence **sequencename**. Elle donne généralement de meilleurs résultats

Liens : *tilt*

sequnsetmag

```
sequnsetmag
```

Réinitialise la calibration de la magnitude et l'étoile de référence pour la séquence. Voir SEQSETMAG

Liens : *seqsetmag*

sequpdate_key

```
sequpdate_key sequencename key value [keycomment]
sequpdate_key sequencename -delete key
```

(suite sur la page suivante)

(suite de la page précédente)

```
sequpdate_key sequencename -modify key newkey
sequpdate_key sequencename -comment comment
```

Same command as UPDATE_KEY but for the sequence **sequencename**. However, this command won't work on SER sequence

Links : *update_key*

seqwiener

```
wiener sequencename [-loadpsf=] [-alpha=]
```

Identique à la commande **WIENER**, mais appliqué à une séquence qui doit être spécifiée en premier argument

Liens : *wiener*

set

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

Met à jour une valeur de paramètre, en utilisant son nom de variable, avec la valeur donnée, ou un ensemble de valeurs en utilisant un fichier ini existant avec l'option **-import=**.
Voir GET pour obtenir des valeurs ou la liste des variables

Liens : *get*

set16bits

```
set16bits
```

Empêche d'enregistrer les images avec 32 bits par canal lors du traitement, utiliser 16 bits à la place

set32bits

```
set32bits
```

Permet d'enregistrer les images en 32 bits par canal lors du traitement

setcompress

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

Définie si les images sont compressé ou pas.

0 signifie qu'il n'y a pas de compression tandis que **1** active la compression.
Si la compression est activée, le type doit être explicitement écrit avec l'option **-type=** ("rice", "gzip1", "gzip2").
Associée à la compression, la valeur de quantification doit être comprise dans la plage [0, 256].

Par exemple, "setcompress 1 -type=rice 16" définit la compression rice avec une quantification de 16

setcpu

```
setcpu number
```

Défini le nombre de fils d'exécution utilisé pour les calculs.

Peut être aussi élevé que le nombre de threads virtuels existant sur le système, qui est le nombre de cœurs du CPU ou le double de ce nombre si l'hyperthreading (Intel HT) est disponible. La valeur par défaut est le nombre maximum de threads disponibles, ce qui signifie que cette valeur doit être utilisée principalement pour limiter la puissance de traitement. Cette valeur est réinitialisée à chaque exécution de Siril. Voir aussi SETMEM

Liens : *setmem*

setext

```
setext extension
```

Définit l'extension utilisée et reconnue par les séquences.

L'argument **extension** peut être "fit", "fts" ou "fits"

setfindstar

```
setfindstar [reset] [-radius=] [-sigma=] [-roundness=] [-focal=] [-pixelsize=] [-
↪convergence=] [ [-gaussian] | [-moffat] ] [-minbeta=] [-relax=on|off] [-minA=] [-
↪maxA=] [-maxR=]
```

Définit le paramètre de détection d'étoiles pour les commandes FINDSTAR et REGISTER.

Si vous ne passez aucun paramètre, vous obtiendrez la liste des valeurs actuelles.
Passer **reset** réinitialise toutes les valeurs aux valeurs par défaut. Vous pouvez alors toujours passer des valeurs après ce mot-clé.

Valeurs paramétrables :

-radius= définit le rayon de la boite de recherche initiale et doit être compris entre 3 et 50.
-sigma= définit le seuil au-dessus du bruit et doit être supérieur ou égal à 0,05.
-roundness= définit la rondeur minimale des étoiles et doit être comprise entre 0 et 0,95. **-maxR** permet de définir une limite supérieure à l'arrondi, pour visualiser uniquement les zones où les étoiles sont significativement allongées, ne pas utiliser pour l'alignement.
-minA et **maxA** définit la limite du minimum et maximum de l'amplitude des étoiles à conserver, normalisé entre 0 et 1.
-focal= définit la longueur focale du télescope.
-pixelsize= définit la taille des pixels du capteur.
-gaussian et **-moffat** configure le modèle de solveur utilisé (Gaussian par défaut).
Si Moffat est sélectionné, **-minbeta=** définit la valeur minimale de beta pour laquelle les étoiles candidates seront acceptées et doit être supérieure ou égale à 0,0 et inférieure à 10,0.
-convergence= définit le nombre d'itération réalisée pour ajuster la PSF et doit être compris entre 1 et 3 (plus tolérant).
-relax= relâche les vérifications qui sont faites sur les étoiles candidats pour évaluer s'ils sont des étoiles ou non, pour permettre aux objets qui n'ont pas la forme d'une étoile d'être quand même acceptés (désactivé par défaut)

Liens : *findstar*, *register*, *psf*

Le seuil de détection des étoiles est calculé comme la médiane de l'image (qui représente généralement le niveau de fond) plus k fois l'écart type, l'écart type étant la déviation standard de l'image (ce qui est une bonne indication de l'amplitude du bruit). Si vous avez beaucoup d'étoiles dans vos images et un bon rapport signal/bruit, il peut être judicieux d'augmenter cette valeur pour accélérer la détection et réduire les faux positifs.

Il est recommandé de tester les valeurs utilisées pour une séquence avec l'interface graphique de Siril, disponible dans la boîte à outils PSF dynamique du menu d'analyse. Cela peut améliorer la qualité de l'alignement en augmentant les paramètres, mais il est également important de pouvoir détecter plusieurs dizaines d'étoiles dans chaque image.

setmag

```
setmag magnitude
```

Étalonne les magnitudes en sélectionnant une étoile et en donnant la magnitude apparente connue.

Tous les calculs de PSF renverront ensuite la magnitude apparente étalonnée, au lieu d'une magnitude apparente relative aux valeurs ADU. Notez que la valeur fournie doit correspondre à la magnitude du filtre d'observation pour être significative.
Pour réinitialiser la constante de magnitude voir UNSETMAG

Liens : *psf* , *unsetmag*

setmem

```
setmem ratio
```

Définit un nouveau rapport entre la mémoire utilisée et la mémoire libre.

La valeur du **Ratio** doit être comprise entre 0.05 et 2, en fonction des autres activités de la machine. Un ratio plus élevé devrait permettre à Siril de traiter plus rapidement, mais un ratio de mémoire utilisée supérieur à 1 nécessitera l'utilisation de la mémoire sur disque, ce qui est très lent et non recommandé, voire parfois non supporté, entraînant un plantage du système. Une quantité fixe de mémoire peut également être définie dans les paramètres génériques, avec SET, au lieu d'un ratio

Liens : *set*

setphot

```
setphot [-inner=20] [-outer=30] [-aperture=10] [-dyn_ratio=4.0] [-gain=2.3] [-min_val=0]␣
↪[-max_val=60000]
```

Obtient ou définit les paramètres de photométrie, principalement utilisés par SEQPSF. Si des arguments sont fournis, ils mettront à jour les paramètres. Aucun n'est obligatoire, tous peuvent être fournis, les valeurs par défaut sont indiquées dans la syntaxe de la commande. À la fin de la commande, la configuration active sera imprimée.

La taille de l'ouverture est dynamique, sauf si elle est forcée. Dans ce cas, la valeur d'ouverture **aperture** des paramètres est utilisée. Si elle est dynamique, le rayon de l'ouverture est défini par le rapport dynamique fourni ("rayon/demi-FWHM").
Les valeurs autorisées pour l'argument **-dyn_ratio** sont comprises dans l'intervalle [1.0, 5.0]. Une valeur en dehors de cette plage fixera automatiquement l'ouverture à la valeur fixe **-aperture**.

Le gain n'est utilisé que s'il n'est pas disponible dans l'en-tête FITS

Liens : *seqpsf*

setref

```
setref sequencename image_number
```

Définit l'image de référence de la séquence donnée en premier argument. **image_number** est le numéro séquentiel de l'image dans la séquence, et non le numéro dans le nom de fichier, commençant à 1

show

```
show [-clear] [{ -list=file.csv | [name] RA Dec }] [-nolog] [-notag]
```

Affiche un point sur l'image chargée et résolue en utilisant le catalogue temporaire d'annotations de l'utilisateur, basé sur ses coordonnées équatoriales. L'option **-clear** efface d'abord ce catalogue et peut être utilisée seule.
Plusieurs points peuvent être transmis en utilisant un fichier CSV avec l'option **-list=** contenant au moins les colonnes ra et dec. Si le fichier transmis contient également un nom de colonne, les noms seront utilisés comme balises dans l'image et listés dans la console, à moins qu'ils ne soient désactivés avec les options **-notag** et **-nolog**.

Cette fonction n'est disponible qu'à partir de l'interface graphique de Siril

spcc

```
spcc [-limitmag=[+-]] [ { -monosensor= [ -rfilter= ] [-gfilter=] [-bfilter=] | -
↪oscsensor= [-oscfilter=] [-osclpf=] } ] [-whiteref=] [ -narrowband [-rwl=] [-gwl=] [-
↪bwl=] [-rbw=] [-gbw=] [-bbw=] ] [-bgtol=lower,upper] [ -atmos [-obsheight=] { [-
↪pressure=] | [-slp=] } ]
```

Exécute la Correction de Couleur par Spectrophotométrie sur l'image chargée et résolue astrométriquement.

La magnitude limite des étoiles est automatiquement calculée à partir de la taille du champ de vision, mais peut être modifiée en passant une valeur +offset ou -offset à **-limitmag=**, ou simplement une valeur positive absolue pour la magnitude limite.
Le catalogue d'étoile utilisé pour SPCC est toujours Gaia DR3 : par défaut le catalogue local Gaia DR3 xp_sampled est utilisé si disponible mais cela peut être remplacé avec **-catalog={gaia | localgaia}**.

Les noms des capteurs et des filtres peuvent être spécifiés à l'aide des options suivantes : **-monosensor=**, **-rfilter=**, **-gfilter=**, **-bfilter=** ou **-oscsensor=**, **-oscfilter=**, **-osclpf=** ; le nom de la référence des blancs peut être spécifié à l'aide de l'option **-whiteref=**. Dans tous les cas, le nom doit être fourni exactement tel qu'il apparaît dans les listes déroulantes de l'outil SPCC. Notez que les noms des capteurs, des filtres et des références blanches peuvent contenir des espaces : dans ce cas, lorsqu'ils sont utilisés comme arguments de la commande **spcc**, l'argument entier doit être placé entre guillemets, par exemple "-whiteref=Average Spiral Galaxy".

Le mode bande étroite peut être sélectionné à l'aide de l'argument **-narrowband**, auquel cas les arguments de filtre précédents sont ignorés et les longueurs d'onde et largeurs de bande du filtre peuvent être fournies à l'aide de **-rwl=**, **-rbw=**, **-gwl=**, **-gbw=**, **-bwl=** et **-bbw=**.

Si l'un des arguments des données spectrales est omis, la valeur utilisée précédemment sera utilisée.

Background reference outlier tolerance can be specified in sigma units using **-bgtol=lower,upper** : these default to -2.8 and +2.0.

La correction atmosphérique peut être appliquée en passant **-atmos**. Dans ce cas, les arguments facultatifs suivants s'appliquent : **-obsheight=** spécifie la hauteur de l'observateur au-dessus du niveau de la mer en mètres (10 par défaut), **-pressure=** spécifie la pression atmosphérique locale sur le site d'observation en hPa, ou **-slp=** spécifie la pression atmosphérique au niveau de la mer en hPa (la pression par défaut est de 1013,25 hPa au niveau de la mer)

spcc_list

```
spcc_list { oscsensor | monosensor | redfilter | greenfilter | bluefilter | oscfilter |␣
↪osclpf | whiteref }
```

Imprime une liste de noms SPCC pouvant être utilisés pour définir des capteurs, des filtres ou des références de blanc à l'aide de la commande **spcc**. Cette commande nécessite un argument pour définir la liste à imprimer : les options sont **oscsensor**, **monosensor**, **redfilter**, **greenfilter**, **bluefilter**, **oscfilter**, **osclpf** ou **whiteref**.
Notez que les noms des capteurs, des filtres et des références de blanc peuvent contenir des espaces : dans ce cas, lorsqu'ils sont utilisés comme arguments de la commande **spcc**, l'argument entier doit être placé entre guillemets, par

exemple "-whiteref=Average Spiral Galaxy"

Liens : *spcc*

split

```
split file1 file2 file3 [-hsl | -hsv | -lab]
```

Divise l'image couleur chargée en trois fichiers distincts (un pour chaque couleur) et les enregistre dans les fichiers **file1**.fit, **file2**.fit et **file3**.fit. Un dernier argument peut être fourni, **-hsl**, **-hsv** ou **lab** pour effectuer une extraction TSL, TSV ou CieLAB. Si aucune option n'est fournie, l'extraction est de type RVB, ce qui signifie qu'aucune conversion n'est effectuée

split_cfa

```
split_cfa
```

Divise l'image CFA chargée en quatre fichiers distincts (un pour chaque canal) et les enregistre dans des fichiers

stack

```
stack seqfilename
stack seqfilename { sum | min | max } [-output_norm] [-out=filename] [-maximize] [-
↪upscale] [-32b]
stack seqfilename { med | median } [-nonorm, -norm=] [-fastnorm] [-rgb_equal] [-output_
↪norm] [-out=filename] [-32b]
stack seqfilename { rej | mean } [rejection type] [sigma_low sigma_high]  [-rejmap[s]] [-
↪nonorm, -norm=] [-fastnorm] [-overlap_norm] [-weight={noise|wfwhm|nbstars|nbstack}] [-
↪feather=] [-rgb_equal] [-output_norm] [-out=filename] [-maximize] [-upscale] [-32b]
```

Emplie la séquence **sequencename** en utilisant les options.

Type de Rejet :
Les types autorisés sont : **sum**, **max**, **min**, **med** (ou **median**) et **rej** (ou **mean**). Si aucun argument autre que le nom de la séquence n'est fourni, l'empilement par somme est supposé.

Empilement avec rejets :

Les types **rej** ou **mean** nécessitent l'utilisation d'arguments supplémentaires pour le type de rejet et les valeurs. Le type de rejet est l'un des **n[one], p[ercentile], s[igma], m[edian], w[insorized], l[inear], g[eneralized], [m]a[d]** pour Percentile, Sigma, Median, Winsorized, Linear-Fit, Generalized Extreme Studentized Deviate Test ou k-MAD clipping. En cas d'omission, la valeur par défaut de Winsorized est utilisée.

Les paramètres **sigma low** et **sigma high** de rejet sont obligatoire sauf si **none** est sélectionné.

En option, des cartes de rejet peuvent être créées, montrant où les pixels ont été rejetés dans une (**-rejmap**) ou deux (**-rejmaps**, pour les rejets "bas" et "hauts") images nouvellement créées.

Normalisation des images d'entrées :

Pour les types d'empilement **med** (ou **median**) et **rej** (ou **mean**), différents types de normalisation sont autorisés : **-norm=add** pour l'additif, **-norm=mul** pour le multiplicatif. Les options **-norm=addscale** et **-norm=mulscale** appliquent la même normalisation mais avec des opérations d'échelle. L'option **-nonorm** permet de désactiver la normalisation. Sinon, la méthode addtive avec mise à l'échelle est appliquée par défaut.

L'option **-fastnorm** spécifie l'utilisation d'estimateurs plus rapides pour la position et l'échelle que l'IKSS par défaut.

-overlap_norm, if passed, will compute normalization coeffcients on images overlaps instead of whole images (allowed only if **-maximize** is passed).

Autres option pour l'empilement avec rejet :

Une pondération peut être appliquée aux images de la séquence à l'aide de l'option **-weight=** suivie de :

noise pour ajouter des poids plus importants aux images ayant un bruit de fond plus faible.

nbstack to weight input images based on how many images were used to create them, useful for live stacking.

nbstars or **wfwhm** to weight input images based on number of stars or wFWHM computed during registration step.

L'option **-feather=** applique un masque de contour progressif sur les bords de chaque image sur la distance (en pixels) indiquée dans l'argument.

Sortie :

Le nom de l'image de résultat pout être définit avec l'option **-out=**. Autrement le nom sera **sequencename**_stacked.fit.

-output_norm applique une normalisation pour replacer le résultat dans l'intervalle [0,1] (seulement pour l'empilement par moyenne ou par médiane).

L'option **-maximize** utilisera les données d'alignement de la séquence pour créer une image stackée qui englobe toutes les images de la séquence (applicable à toutes les méthodes sauf l'empilement médian).

L'option **-upscale** augmentera la résolution de la séquence d'un facteur 2 avant l'empilement à l'aide des données d'alignement (applicable à toutes les méthodes sauf l'empilement médian).

-rgb_equal will use normalization to equalize color image backgrounds, useful if PCC/SPCC or unlinked AUTOSTRETCH will not be used.

-32b remplacera la profondeur de bits définie dans les préférences et enregistrera l'image empilée en 32b.

Filtrage des images :

Les images à empiler peuvent être sélectionnées avec certain filtres, tel que une sélection manuel ou la meilleur FWHM, avec d'autres option **-filter-***.

Links : *pcc*, *spcc*, *autostretch*

```
[-filter-fwhm=value[%|k]] [-filter-wfwhm=value[%|k]] [-filter-round=value[%|k]] [-filter-
```

(suite sur la page suivante)

(suite de la page précédente)

```
↪bkg=value[%|k]]
[-filter-nbstars=value[%|k]] [-filter-quality=value[%|k]] [-filter-incl[uded]]
```

Les meilleurs images de la séquence peuvent être empilées en utilisant les arguments de filtrage. Chacun de ces argument peu supprimer les mauvaises images sur la base des propriétés que leur nom contient, récupéré depuis les données d'alignement, avec l'un des trois types de valeur d'argument :

- 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.

Il est également possible d'utiliser des images sélectionnées manuellement, soit précédemment à partir de l'interface graphique utilisateur (GUI), soit avec les commandes select ou unselect, en utilisant l'argument **-filter-included**.

stackall

```
stackall
stackall { sum | min | max } [-maximize] [-upscale] [-32b]
stackall { med | median } [-nonorm, norm=] [-32b]
stackall { rej | mean } [rejection type] [sigma_low sigma_high] [-nonorm, norm=] [-
↪overlap_norm] [-weight={noise|wfwhm|nbstars|nbstack}] [-feather=] [-rgb_equal] [-
↪out=filename] [-maximize] [-upscale] [-32b]
```

Ouvre toutes les séquences dans le répertoire actuel et les empile avec le type d'empilement et le filtrage éventuellement spécifiés ou avec l'empilement par somme. Voir la commande STACK pour la description des options

Liens : *stack*

starnet

```
starnet [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

Appelle StarNet pour supprimer les étoiles de l'image chargée.

StarNet est un programme externe, sans affiliation avec Siril, et doit être installé correctement avant la première utilisation de cette commande, avec le chemin d'accès à son installation en version CLI correctement défini dans Préférences / Divers.

L'image sans étoile (starless) est chargée à la fin et une image de masque d'étoile est créée dans le répertoire de travail à moins que le paramètre facultatif **-nostarmask** ne soit fourni.

En option, des paramètres peuvent être passés à la commande :
- The option **-stretch** is for use with linear images and will apply a pre-stretch before running StarNet and the inverse stretch to the generated starless and starmask images.
- To improve star removal on images with very tight stars, the parameter **-upscale** may be provided. This will upsample the image by a factor of 2 prior to StarNet processing and rescale it to the original size afterwards, at the expense of more processing time.
- The optional parameter **-stride=value** may be provided, however the author of StarNet *strongly* recommends that the default stride of 256 be used

start_ls

```
start_ls [-dark=filename] [-flat=filename] [-rotate] [-32bits]
```

Initialise une session de livestacking, en utilisant les fichiers de calibration optionnels et attend que les fichiers d'entrée soient fournis par la commande LIVESTACK jusqu'à ce que STOP_LS soit appelé. Le traitement par défaut utilise l'alignement par translation seulement et le traitement 16 bits parce qu'il est plus rapide, il peut être changé en rotation avec **-rotate** et **-32bits**

Notez que les commandes de live stacking mettent Siril dans un état dans lequel il n'est pas capable de traiter d'autres commandes. Après START_LS, seuls LIVESTACK, STOP_LS et EXIT peuvent être appelés jusqu'à ce que STOP_LS soit appelé pour remettre Siril dans son état normal, sans livestacking

Liens : *livestack*, *stop_ls*, *exit*

stat

```
stat [-cfa] [main]
```

Renvoie les statistiques de l'image courante, la liste de base par défaut ou la liste principale si **main** est passé. Si une sélection est faite, les statistiques sont calculées dans la sélection. Si **-cfa** est passé et que l'image est CFA, les statistiques sont faites sur les extractions par filtre

stop_ls

```
stop_ls
```

Arrête la session de livestacking. Uniquement possible après START_LS

Liens : *start_ls*

subsky

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-
↪existing]
```

Calcule un fond de ciel synthétique en utilisant soit le modèle de fonction polynomiale de **degree** degrés, soit le modèle RBF (si **-rbf** est fourni à la place) et le soustrait de l'image.

Le nombre d'échantillons par ligne horizontale et la tolérance pour exclure les zones plus claires peuvent être ajustés avec les arguments optionnels. La tolérance est exprimée en unités MAD : médiane + tolérance * mad.

La diffusion d'erreur, nécessaire pour les gradients peu dynamiques, peut être activé avec **-dither**.

Pour RBF, le paramètre de lissage supplémentaire est également disponible. Pour utiliser des échantillons d'arrière-plan préexistants (par exemple, si vous avez défini des échantillons d'arrière-plan à l'aide d'un script Python), l'argument **-existing** doit être utilisé

synthstar

```
synthstar
```

Corrige les étoiles imparfaites de l'image chargée. Quelle que soit l'importance de la coma, de la dérive ou de toute autre distorsion de vos étoiles, si la routine de recherche d'étoiles de Siril peut les détecter, synthstar les corrigera. Pour une utilisation intensive, vous pouvez souhaiter détecter manuellement toutes les étoiles que vous souhaitez corriger. Cela peut être fait en utilisant la commande findstar de la console ou la boîte de dialogue PSF dynamique. Si vous n'avez pas lancé la détection d'étoiles, elle sera lancée automatiquement avec les paramètres par défaut.

Pour de meilleurs résultats, synthstar doit être exécuté avant l'étirement.

La sortie de synthstar est un masque d'étoile synthétique entièrement corrigé comprenant des PSF d'étoiles parfaitement rondes (profils Moffat ou gaussiens en fonction de la saturation de l'étoile) calculés pour correspondre à l'intensité, à la FWHM, à la teinte et à la saturation mesurées pour chaque étoile détectée dans l'image d'entrée. Cela peut ensuite être recombiné avec l'image sans étoiles pour produire une image avec des étoiles parfaites.

Aucun paramètre n'est requis pour cette commande

Liens : *psf*

threshlo

```
threshlo level
```

Remplace les valeurs inférieures à **level** dans l'image chargée par **level**

threshhi

```
threshi level
```

Remplace les valeurs supérieures à **level** dans l'image chargée par **level**

thresh

```
thresh lo hi
```

Remplace les valeurs inférieures à **level** dans l'image chargée par **level**

tilt

```
tilt [clear]
```

Calcule le tilt du capteur comme la différence FWHM entre les valeurs moyennes tronquées des meilleurs et des pires coins. L'option **clear** permet d'effacer le dessin

trixel

```
trixel [-p]
```

Pour les développeurs.

Sans argument, liste tous les trixels de niveau 3 visibles dans l'image résolue en astrométrie. Les étoiles de chaque trixel peuvent ensuite être affichées avec la commande CONESEARCH en utilisant **-trix=** suivi d'un numéro de trixel visible

Avec l'argument **-p**, imprime toutes les étoiles valides des 512 trixels de niveau 3 dans le fichier "trixels.csv"

Links : *conesearch*

unclipstars

```
unclipstars
```

Reprofile les étoiles coupées de l'image chargée pour les désaturer, en mettant à l'échelle la sortie de manière à ce que toutes les valeurs de pixels soient <= 1,0

unpurple

```
unpurple [-starmask] [-blue=value] [-thresh=value]
```

Applique un filtre cosmétique pour réduire les effets de franges violettes sur les étoiles.

If the **-starmask** parameter is given, a star mask will be used to identify areas of the image to affect. If a Dynamic PSF has already been run, this will be used for the starmask, otherwise one will be created automatically. The **-mod=** parameter should be given a value somewhere around 0.14 to reduce the amount of purple. The **-thresh=** will specify the size modifier for each star in the starmask and should be large enough to cause the stars to be entirely processed without remaining purple fringing. The value should between 0 and 1, typically around 0.5.

If the **-starmask** parameter is not given, the purple reduction will be applied across the entire image for any purple pixels with a luminance value higher than the given **-thresh=**. In this case, the **-thresh=** value should be reasonably low. This mode is useful for starmasks or other images without nebula or galaxy

Liens : *psf*

unselect

```
unselect sequencename from to
```

Permet une sélection en masse des images dans la séquence **sequencename** (de **from** à **to** inclue). Voir SELECT

Liens : *select*

unsetmag

```
unsetmag
```

Remet l'étalonnage de la magnitude à 0. Voir SETMAG

Liens : *setmag*

unsharp

```
unsharp sigma multi
```

Applique un masque de netteté, en fait une image filtrée gaussienne avec sigma **sigma** et un mélange avec le paramètre **amount** utilisé comme suit : out = in * (1 + amount) + filtré * (-amount).

Voir aussi GAUSS, la même chose sans mélange

Liens : *gauss*

update_key

```
update_key key value [keycomment]
update_key -delete key
```

(suite sur la page suivante)

(suite de la page précédente)

```
update_key -modify key newkey
update_key -comment comment
```

Updates FITS keyword. Please note that the validity of **value** is not checked. This verification is the responsibility of the user. It is also possible to delete a key with the **-delete** option in front of the name of the key to be deleted, or to modify the key with the **-modify** option. The latter must be followed by the key to be modified and the new key name. Finally, the **-comment** option, followed by text, adds a comment to the FITS header. Please note that any text containing spaces must be enclosed in double quotation marks

visu

```
visu low high
```

Affiche l'image chargée avec **low** et **high** comme seuils bas et haut, interface graphique uniquement

wavelet

```
wavelet nbr_layers type
```

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

Voir aussi EXTRACT

Liens : *wrecons*, *extract*

wiener

```
wiener [-loadpsf=] [-alpha=]
```

Restore une image en utilisant la méthode de déconvolution de Wiener.

En option, une PSF créée par MAKEPSF peut être chargée en utilisant l'argument **-loadpsf=filename**.

Le paramètre **-alpha=** fournit le facteur de régularisation modélisé par le bruit Gaussien

Liens : *psf*, *makepsf*

wrecons

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

Reconstruit l'image actuelle à partir des couches précédemment calculées avec les ondelettes et pondérées avec les coefficients **c1**, **c2**, ..., **cn** en fonction du nombre de couches utilisées pour la transformation en ondelettes, après l'utilisation de WAVELET

Liens : *wavelet*

CHAPITRE 25

Sirilpy Python Module API 1.0.25 Reference

The sirilpy python module supports communication with a running Siril instance. It can request representations of the current loaded image plus its metadata, including details of detected stars, as well as the current loaded sequence and most frame metadata.

Cette documentation est généré automatique depuis la version 1.0.25 du module python.

It can also run Siril commands using the `SirilInterface.cmd()` method, and the intent is to provide a capable interface for writing advanced scripts for Siril to a level not possible with the previous simple script files.

For example, scripts can now have PyQt6 or TKinter front ends using the PyQt6 or tkinter and ttkthemes modules, and they can utilise a lot of the ecosystem of Python modules including numpy, scipy, pillow and many more.

> **Note**
>
> Il existe certaines restrictions concernant les modules qui nécessitent l'installation de paquets binaires système pour fonctionner.

In the initial module release, most methods relating to the image or sequence loaded in Siril are read-only. The intent is that the parameters of the loaded image or sequence can be obtained and used as inputs to scripts, for example for calculations or input to conditionals, but the existing mature Siril command set should in most cases be used to act on the loaded image. Thus header keywords can be set using `cmd("update_key", "key", "value")`, and most built-in image operations can be carried out using the appropriate Siril command. The main exception to the rule of python methods providing read-only access is the `set_image_pixeldata()` method, which allows for setting the pixel data in the loaded image from a numpy array. This means that new pixel processing algorithms can be added using python, initially getting the pixel data from the loaded image using `get_image_pixeldata()` and setting it on completion using `set_image_pixeldata()`. Similar functions are available for getting and setting pixel data from sequence frames.

25.1 Dépendances

The sirilpy module depends on the following :

- numpy >= 1.20.0
- packaging >= 21.0
- pywin32 >= 300 (seulement sur Windows)

25.2 Guide de codage des Script Siril

- Unlike most python environments, Siril scripts run directly from Siril and the end user may not typically know how to access or how to use the python venv from outside Siril. This means that some tasks that are fairly trivial in typical python scenarios, such as installing packages, become harder. The user cannot be expected to access the command line and install packages themselves using `python3 -m pip install`.
  - Le module fournit donc la méthode «`ensure_installed()`. Celle-ci utilise pip pour s'assurer que les modules sont installés afin qu'ils puissent être importés.
- Siril scripts should **always** provide the author's name, copyright details / license information and contact details such as a YouTube channel, website or forum where they can be contacted in regard to their script. This is advisory for independently distributed scripts and mandatory for any scripts that are submitted to the scripts repository. If you write a script, you are responsible for supporting it.
- Au fur et à mesure que le répertoire s'enrichit et que l'API évolue, tous les scripts publiés ne seront pas nécessairement toujours compatibles avec toutes les versions de Siril qui seront utilisé :
  - If your script uses Siril commands, you should use the `requires` Siril command. This can be called directly in a Siril Script File, or it can be called from a Python script using `SirilInterface.cmd("requires", "min_version", {"max_version"})` (max_version is optional) but can be used to ensure scripts designed for older versions of Siril no longer show as applicable to newer versions if the command syntax has changed.
  - If your script uses any features of the Siril python module added since the initial release, you should call the `sirilpy.check_module_version()` method to ensure the installed version meets the requriements of your script. The versions at which features were added will be listed in the API documentation for all features added after the initial public release.
  - The code that populates the Siril view of the scripts repository will automatically filter out scripts whose Siril version or python module version requirements are not met.
  - If new classes or methods are added to the module after its initial public release the version at which they were introduced will be annotated in the docstring and the online documentation generated from it.
- Siril targets Linux, Windows and MacOS. Script writers are encouraged, where possible, to ensure that their scripts run correctly on all three OSes.

25.3 Connexion Sirilpy

This submodule provides the main SirilInterface class used for communication between Siril and the python script. All its members are made available at the module root level, there is no need to import connection separately. Connection module for Siril, providing the ability to connect to a running Siril instance and communicate with it. Includes an extensive range of methods that can be used to get and set data from / to Siril.

class sirilpy.connection.**SirilInterface**

Bases : object

SirilInterface is the main class providing an interface to a running Siril instance and access to methods to interact with it through Siril's inbuilt command system and accessing image and sequence data.

analyse_image_from_file(*filepath*)

Request Siril to load an image from a file and analyse it. This method does not change the image currently loaded in Siril. Any image format supported by Siril is supported. An ImageAnalysis object is returned, containing parameters that may be used to assess the quality of an image for use in culling.

Paramètres

filepath (str) -- Chaîne de caractères indiquant le chemin d'accès au fichier image à charger.

Type renvoyé

ImageAnalysis

Renvoie

Objet ImageAnalysis contenant les indicateurs de qualité de l'image analysée.

Lève

— **FileNotFoundError** -- si le fichier indiqué n'existe pas,
— *DataError* -- en cas de réception de données erronées,
— *SirilError* -- si une erreur est survenue.

clear_image_bgsamples()

Clears all background sample points from the image.

clear_undo_history()

Efface l'historique des annulations de l'image, s'il y a un historique à effacer

cmd(**args*)

Send a command to Siril to be executed. The range of available commands can be found by checking the online documentation. The command and its arguments are provided as a list of strings.

Paramètres

***args** (str) -- Variable number of string arguments to be combined into a command

Lève

— *DataError* -- Si aucune réponse (ou une réponse incorrecte) a été reçue,
— *CommandError* -- Si la commande renvoie un code d'erreur,
— *SirilError* -- Si une autre erreur est survenue durant l'exécution.

Exemple

```
siril.cmd("ght", "-D=0.5", "-b=2.0")
```

command_error_message(*status_code*)

Provides a string describing the return status from a Siril command.

Paramètres

status_code (*CommandStatus*) -- The status code returned by the Siril command handler, or by the CommandError exception.

Renvoie

A string providing a description of the error code returned by a Siril command, for use in exception handling.

Type renvoyé
str

`confirm_messagebox`(*title*, *message*, *confirm_label*)

Créez une boîte de dialogue modale de confirmation dans Siril et attendez la réponse.

Paramètres
— **title** (str) -- Le titre à afficher dans la boîte de message (256 caractères maximum)
— **message** (str) -- Le message à afficher dans la boîte de dialogue (1 021 caractères maximum)
— **confirm_label** (str) -- Le libellé à afficher sur le bouton de confirmation de la boîte de dialogue (OK, Oui, Confirmer, etc.) (24 caractères maximum)

Renvoie
True si l'on a cliqué sur le bouton de confirmation de la boîte de dialogue, False dans le cas contraire

Type renvoyé
bool

Lève
— *DataError* -- si aucune réponse n'est reçu,
— *SirilError* -- si une autre erreur est survenue.

`connect`()

Establish a connection to Siril based on the pipe or socket path.

Type renvoyé
bool

Renvoie
True en cas de succès

Lève
SirilConnectionError -- si une erreur de connexion s'est produite

`create_new_seq`(*seq_root*)

Creates a new .seq file with all images named seq_rootXXXXX.ext located in the current home folder. If a sequence with the same name is already loaded in Siril, it will not be recreated. This only works for FITS files, not FITSEQ nor SER. The newly created sequence is not loaded in Siril.

Paramètres
seq_root (str) -- Le nom racine de la séquence à créer.

Renvoie
True if the sequence was successfully created, False otherwise.

Type renvoyé
bool

Lève
SirilError -- si une erreur est survenue.

`disconnect`()

Closes the established socket or pipe connection. Note there is not usually any need to close this unless for some reason you wish to close a connection and subsequently reopen another one. This method is automatically called at script termination using an `atexit` handler so there is no need to do so manually. Calling this method will reset the progress bar.

Lève
SirilConnectionError -- si la connexion ne peut pas être fermée.

error_messagebox(*my_string*, *modal=False*)

Send an error message to Siril. The maximum message length is 1022 bytes : longer messages will be truncated (but this is more than enough for an error message box). Note that the error message box is not modal by default : this is intended for displaying an error message more prominently than using the Siril log prior to quitting the application.

Paramètres
- **my_string** (str) -- Le message à afficher dans la boîte de message d'erreur
- **modal** (Optional[bool]) -- Sets whether or not the message box should be modal and wait for completion or non-modal and allow the script to continue execution. Note that although a modal message box will block execution of the script, if a TKinter main loop is running events will continue to queue up, so if the message box is triggered by clicking a button then the user may click it while the message box is shown and trigger a second message box which will display immediately the first one is closed.

Renvoie
Vrai si l'erreur a été affichée avec succès, faux dans le cas contraire

Type renvoyé
bool

Lève
SirilError -- si une erreur est survenue.

get_image(*with_pixels=True*, *preview=False*)

Request a copy of the current image open in Siril. Requires a single image to be loaded.

Paramètres
- **with_pixels** (Optional[bool]) -- booléen facultatif indiquant s'il faut récupérer les données pixel sous forme d'un NumPy array ou uniquement les métadonnées de l'image. La valeur par défaut est True
- **preview** (Optional[bool]) -- booléen facultatif indiquant s'il faut obtenir les données pixel sous forme d'aperçu (i.e. des données étirées automatiquement sur 8 bits) ou sous forme de données image réelles. La valeur par défaut est False (i.e. des données image réelles)

Type renvoyé
Optional[*FFit*]

Renvoie
Objet FFit contenant les métadonnées de l'image et (éventuellement) les données pixel, ou None si une erreur s'est produite

Lève
- *NoImageError* -- si aucune image n'est chargée dans Siril
- *SirilError* -- si une erreur de décodage se produit

get_image_bgsamples()

Request background samples data from Siril.

Type renvoyé
Optional[List[*BGSample*]]

Renvoie
List of BGSamples background samples, with each set of coordinates expressed as a tuple[float, float], or None if no background samples have been set.

Lève
- *NoImageError* -- Si aucune image n'est actuellement chargée,
- *DataError* -- à la réception de mauvaise données,
- *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données,

`get_image_filename`()

Request the filename of the loaded image from Siril. Requires a single image to be loaded.

Type renvoyé
`Optional[str]`

Renvoie
Le nom de fichier comme une chaîne de caractère.

Lève
— *NoImageError* -- si aucune image n'est chargé,
— *SirilError* -- si une erreur de décodage se produit.

`get_image_fits_header`(*return_as='str'*)

Récupérer l'entête FITS complet de l'image actuellement chargée dans Siril. Ceci nécessite qu'une seule image soit chargée.

Paramètres
return_as -- Chaîne facultative spécifiant le format de l'en-tête renvoyé. Peut prendre la valeur 'str' pour une chaîne de caractères ou 'dict' pour un dictionnaire.

Renvoie
The image FITS header as a string, or None if there is no header. dict : The image FITS header as a dictionary, or None if there is no header. None : If the header is empty or not available.

Type renvoyé
str

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données,

`get_image_history`()

Retrieve history entries in the FITS header of the current loaded Siril image using shared memory. Requires a single image to be loaded.

Paramètres
none.

Renvoie
Les entrées HISTORY dans l'entête FITS sous forme de liste de chaînes, ou None s'il n'y a pas de mots-clés HISTORY.

Type renvoyé
list

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données.

`get_image_iccprofile`()

Retrieve the ICC profile of the current Siril image using shared memory. Requires a single image to be loaded.

Args : none.

Renvoie
The image ICC profile as a byte array, or None if the current image has no ICC profile.

Type renvoyé
bytes

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— *SirilError* -- Si une autre erreur se produit.

`get_image_keywords`()

Request FITS keywords data from Siril as a FKeywords object. Requires a single image to be loaded.

Type renvoyé
Optional[*FKeywords*]

Renvoie
Objet FKeywords contenant les mots-clés FITS, ou None si une erreur s'est produite

Lève
SirilError -- si une erreur de décodage se produit.

`get_image_pixeldata`(*shape=None*, *preview=False*, *linked=False*)

Retrieves the pixel data from the image currently loaded in Siril.

Paramètres
— **shape** (Optional[list[int]]) -- Optional list of [x, y, w, h] specifying the region to retrieve. If provided, gets pixeldata for just that region. If None, gets pixeldata for the entire image.
— **preview** (Optional[bool]) -- booléen facultatif indiquant s'il faut obtenir les données pixel sous forme d'aperçu (i.e. des données étirées automatiquement sur 8 bits) ou sous forme de données image réelles. La valeur par défaut est False (i.e. des données image réelles)
— **linked** (Optional[bool]) -- optional bool specifying whether the autostretch preview should be linked or unlinked. If preview == False then this option is ignored.

Renvoie
Les données d'image en format numpy array

Type renvoyé
numpy.ndarray

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— **ValueError** -- Si une taille invalide est fournie,
— *DataError* -- si le tableau ne peut pas être redimensionné à la bonne dimension,
— *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données pixel,

`get_image_shape`()

Demande la taille de l'image à Siril.

Type renvoyé
Optional[Tuple[int, int, int]]

Renvoie
A tuple (channels, height, width) representing the shape of the image, or None if no image shape is available to return.

Raises : SirilError : if an error occurred.

`get_image_stars`(*channel=None*)

Request star model PSF data from Siril.

Paramètres
channel (Optional[int]) -- Optional int specifying the channel to retrieve from. If provided 0 = Red / Mono, 1 = Green, 2 = Blue. If the channel is omitted the default behavior will be used : channel 0 for mono images, channel 1 (green) for color images. **channel requires sirilpy v1.0.8 or higher.**

Type renvoyé
List[*PSFStar*]

Renvoie
List of PSFStar objects containing the star data, or None if no stars can be found. If stars have already been detected using the *findstar* command then this list will be returned, otherwise automatic star detection will be attempted with the current star finder settings.

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— **ValueError** -- If an invalid channel is provided,
— *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données,

get_image_stats(*channel*)

Demander à Siril les statistiques relatives à une image pour un canal spécifique. Nécessite le chargement d'une seule image.

Paramètres
channel (int) -- Nombre entier indiquant le canal pour lequel obtenir les statistiques (généralement 0, 1 ou 2)

Type renvoyé
Optional[*ImageStats*]

Renvoie
ImageStats object containing the statistics, or None if no stats are available for the selected channel

Lève
— *NoImageError* -- si aucune image n'est chargé,
— *SirilError* -- si une erreur est survenue.

get_image_unknown_keys()

Retrieve the unknown key in a FITS header of the current loaded Siril image using shared memory. Requires a single image to be loaded.

Paramètres
none.

Renvoie
Les clés inconnues sous forme de chaîne, ou None s'il n'y a pas de clés inconnues.

Type renvoyé
bytes

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données.

get_selection_star(*shape=None*, *channel=None*, *assume_centred=False*)

Retrieves a PSFStar star model from the current selection in Siril. Only a single PSFStar is returned : if there are more than one in the selection, the first one identified by Siril's internal star detection algorithm is returned. **Update** : from sirilpy 1.0.4 this method uses Siril's photometry functions to try to provide photometrically accurate values for PSFStar.mag, PSFStar.s_mag and PSFStar.SNR. If photometry succeeded and no saturated pixels were detected then PSFStar.phot_is_valid will be True, otherwise it will be False. :type shape : Optional[list[int]] :param shape : Optional list of [x, y, w, h] specifying the selection to

retrieve from. w x h must not exceed 300 px x 300 px. If provided, looks for a star in the specified selection If None, looks for a star in the selection already made in Siril, if one is made.

Paramètres
— **channel** (Optional[int]) -- Optional int specifying the channel to retrieve from. If provided 0 = Red / Mono, 1 = Green, 2 = Blue. If the channel is omitted the current viewport will be used if in GUI mode, or if not in GUI mode the method will fall back to channel 0
— **assume_centred** (bool) -- Optional bool specifying whether to assume the star is already centred in the selection. Defaults to False.

Renvoie

l'objet PSFStar représentant le modèle d'étoile, ou None si
aucune étoile n'a été détectée dans la sélection.

Type renvoyé
PSFStar

Lève
— **ValueError** -- Si une taille invalide est fournie,
— ***NoImageError*** -- Si aucune image n'est chargé,
— ***SirilConnectionError*** -- Si une erreur de communication se produit,
— ***SirilError*** -- Si une autre erreur s'est produite pendant l'exécution.

get_selection_stats(*shape=None*, *channel=None*)

Récupère les statistiques relatives à la sélection actuelle dans Siril. Nécessite le chargement d'une image unique ou d'une séquence.

Paramètres
— **shape** (Optional[list[int]]) -- Optional list of [x, y, w, h] specifying the selection to retrieve from. If provided, looks for a star in the specified selection If None, looks for a star in the selection already made in Siril, if one is made.
— **channel** (Optional[int]) -- Optional int specifying the channel to retrieve from. If provided 0 = Red / Mono, 1 = Green, 2 = Blue. If the channel is omitted the current viewport will be used if in GUI mode, or if not in GUI mode the method will fall back to channel 0

Renvoie
the ImageStats object representing the selection statistics.

Type renvoyé
ImageStats

Lève
— ***SirilError*** -- If an error occurred during processing,
— **ValueError** -- Si une forme non valide est fournie.

get_seq()

Demander les métadonnées de la séquence actuellement chargée dans Siril.

Type renvoyé
Optional[*Sequence*]

Renvoie
Objet de séquence contenant les métadonnées de la séquence actuelle, ou None si une erreur s'est produite

Lève
— ***NoSequenceError*** -- si aucune séquence n'est chargée dans Siril,
— ***SirilError*** -- si une erreur de décodage se produit.

get_seq_distodata(*channel*)

Request sequence distortion data from Siril

channel : Integer specifying which channel to get registration data for (typically 0, 1, or 2)

Type renvoyé
Optional[*DistoData*]

Renvoie
DistoData object containing the channel distortion parameters, or None if an error occurred

Lève
— ***NoSequenceError*** -- si aucune séquence n'est chargée dans Siril,
— ***SirilError*** -- si une erreur de décodage se produit.

get_seq_frame(*frame*, *with_pixels=True*, *preview=False*, *linked=False*)

Request sequence frame as a FFit from Siril. The keywords, statistics, header and other metadata are always returned : if an ICC profile is present, this will also be populated in the resulting FFit, and optionally the pixel data can also be returned.

Paramètres
- **frame** (int) -- Integer specifying which frame in the sequence to retrieve data for (between 0 and Sequence.number - 1). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **with_pixels** (Optional[bool]) -- bool specifying whether or not to return the pixel data for the frame (default is True).
- **preview** (Optional[bool]) -- bool specifying whether or not to return the real pixel data or an autostretched uint8_t preview version. Only has an effect in conjunction with with_pixels = True

Type renvoyé
Optional[*FFit*]

Renvoie
FFit object containing the frame data

Lève
- *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
- *DataError* -- en cas de réception de données erronées,
- *SirilError* -- si une erreur est survenue.

get_seq_frame_filename(*frame*)

Request the filename of the specified frame of the loaded sequence from Siril. Requires a sequence to be loaded.

Paramètres
frame (*int*) -- Specifies the frame index. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.

Type renvoyé
Optional[str]

Renvoie
Le nom de fichier comme une chaîne de caractère.

Lève
- *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
- *SirilError* -- si une erreur de décodage se produit.

get_seq_frame_header(*frame*, *return_as='str'*)

Retrieve the full FITS header of an image from the sequence loaded in Siril.

Paramètres
- **frame** (int) -- Integer specifying which frame in the sequence to retrieve data for (between 0 and Sequence.number - 1). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **return_as** -- Chaîne facultative spécifiant le format de l'en-tête renvoyé. Peut prendre la valeur 'str' pour une chaîne de caractères ou 'dict' pour un dictionnaire.

Renvoie
The image FITS header as a string, or None if there is no header. dict : The image FITS header as a dictionary, or None if there is no header. None : If the header is empty or not available.

Type renvoyé
str

Lève
- *NoSequenceError* -- Si aucune séquence n'est actuellement chargé,
- *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données,

get_seq_frame_pixeldata(*frame*, *shape=None*, *preview=False*, *linked=False*)

Retrieves the pixel data from a frame in the sequence currently loaded in Siril.

Paramètres

— **frame** (int) -- selects the frame to retrieve pixel data from. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
— **shape** (Optional[List[int]]) -- Optional list of [x, y, w, h] specifying the region to retrieve. If provided, gets pixeldata for just that region. If None, gets pixeldata for the entire image.
— **preview** (Optional[bool]) -- optional bool specifying whether to get pixeldata as a preview (i.e. 8-bit autostretched data) or as real image data. Defaults to False (i.e. real image data).
— **linked** (Optional[bool]) -- optional bool specifying whether the autostretched preview should be linked or unlinked. This option is ignored if preview is not True

Renvoie
Les données d'image en format numpy array

Type renvoyé
numpy.ndarray

Lève
— **ValueError** -- Si une taille invalide est fournie,
— *DataError* -- si le tableau ne peut pas être redimensionné à la bonne dimension,
— *SirilError* -- Pour les autres erreurs survenant lors de la récupération des données des pixels.

`get_seq_imgdata`(*frame*)

Demander les métadonnées des images de la séquence à Siril.

Paramètres
frame (int) -- Integer specifying which frame in the sequence to get image metadata for (between 0 and Sequence.number). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.

Type renvoyé
Optional[*ImgData*]

Renvoie
Objet ImgData contenant les métadonnées de l'image, ou None si une erreur s'est produite

Lève
— *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
— *SirilError* -- si une erreur de décodage se produit.

`get_seq_regdata`(*frame*, *channel*)

Request sequence frame registration data from Siril.

Paramètres
— **frame** (int) -- Integer specifying which frame in the sequence to get registration data for (between 0 and Sequence.number). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
— **channel** (int) -- Integer specifying which channel to get registration data for (typically 0, 1, or 2)

Type renvoyé
Optional[*RegData*]

Renvoie
RegData object containing the registration data, or None if no registration data is available for the specified frame and channel

Lève
— *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
— *SirilError* -- si une erreur de décodage se produit.

`get_seq_stats`(*frame*, *channel*)

Demander les statistiques sur les images de la séquence à Siril.

Paramètres
- **frame** (`int`) -- Integer specifying which frame in the sequence to get statistics data for (between 0 and Sequence.number). This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
- **channel** (`int`) -- Nombre entier indiquant le canal pour lequel obtenir les statistiques (généralement 0, 1 ou 2)

Type renvoyé
`Optional[ImageStats]`

Renvoie
Objet ImageStats contenant les statistiques, ou None si une erreur s'est produite

Lève
- *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
- *SirilError* -- si une erreur de décodage se produit.

`get_siril_active_vport`()

Demande à Siril d'indiquer la fenêtre d'affichage active.

Renvoie
- sirilpy.SirilVport.RED / sirilpy.SirilVport.MONO
- sirilpy.SirilVport.GREEN,
- sirilpy.SirilVport.BLUE,
- sirilpy.SirilVport.RGB

Note that RED and MONO share the same IntEnum value, so there is no difference between a test for one and the other ; the two enum labels are provided solely to aid code legibility.

Type renvoyé
A SirilVport representing the active vport

Lève
- *DataError* -- si aucune réponse n'est reçue ou si la réponse n'est pas valide,
- *SirilError* -- si une erreur est survenue.

`get_siril_config`(*group*, *key*)

Demandez une valeur de configuration à Siril.

Paramètres
- **group** (`str`) -- Nom du groupe de configuration,
- **key** (`str`) -- Configuration key name within the group (Available values for group and key can be determined using the "get -A" command)

Type renvoyé
`Union[bool, int, float, str, List[str], None]`

Renvoie
La valeur de configuration avec le type Python correspondant, ou None si une erreur s'est produite.

Lève
- *DataError* -- si un type de configuration inconnu est détecté,
- *SirilError* -- si une erreur s'est produite lors de la récupération de la valeur de configuration demandée

`get_siril_configdir`()

Demander le répertoire de configuration utilisateur utilisé par Siril.

Type renvoyé
`str`

Renvoie
Le répertoire de configuration de l'utilisateur sous forme de chaîne de caractères.

Lève
— *DataError* -- si aucune réponse n'est reçue,
— *SirilError* -- pour toutes les autres erreurs.

`get_siril_display_iccprofile()`

Récupérer le profil ICC de l'écran de Siril.

Args : none.

Renvoie
Le profil ICC de l'affichage sous forme de tableau d'octets, ou None si Siril fonctionne en mode sans interface graphique.

Type renvoyé
bytes

Lève
— *NoImageError* -- Si aucune image n'est actuellement chargée,
— *SirilError* -- Si une autre erreur se produit.

`get_siril_log()`

Récupérer l'intégralité des logs Siril sous forme de chaîne de texte.

Renvoie
Le texte des logs Siril.

Type renvoyé
str

Lève
SirilError -- En cas d'erreurs lors de la récupération des données,

`get_siril_panzoom()`

Request the pan and zoom state from Siril.

Type renvoyé
`Tuple[float, float, float]`

Renvoie
A tuple (x offset, y offset, zoomlevel) representing the current image pan and zoom state.

Lève
SirilError -- si une erreur est survenue.

`get_siril_selection()`

Demandez la sélection d'images à Siril.

Type renvoyé
`Optional[Tuple[int, int, int, int]]`

Renvoie
A tuple (x, y, height, width) representing the current selection, or None if no selection is made.

Lève
SirilError -- si une erreur est survenue.

`get_siril_slider_state(float_range=False)`

Request the display slider state from Siril.

Type renvoyé
`Tuple[int, int, SlidersMode]`

Renvoie
A tuple (min, max, slider mode) representing the current slider state.

Lève
SirilError -- si une erreur est survenue.

get_siril_stf()

Request the Screen Transfer Function in use in Siril.

Type renvoyé
STFType

Renvoie
A STFType representing the current STF state.

Lève
SirilError -- si une erreur est survenue.

get_siril_stf_linked()

Determine whether the AutoStretch STF is configured to be linked or not

Type renvoyé
bool

Renvoie
A bool representing the current STF channel-linked state.

Lève
SirilError -- si une erreur est survenue.

get_siril_systemdatadir()

Request the system data directory used by Siril.

Type renvoyé
Optional[str]

Renvoie
Le répertoire de données du système sous forme de chaîne de caractères.

Lève
— *DataError* -- si aucune réponse n'est reçue,
— *SirilError* -- pour toutes les autres erreurs.

get_siril_userdatadir()

Request the user data directory used by Siril.

Type renvoyé
str

Renvoie
Le répertoire des données utilisateur sous forme de chaîne de caractères.

Lève
— *DataError* -- si aucune réponse n'est reçue,
— *SirilError* -- pour toutes les autres erreurs.

get_siril_wd()

Request the current working directory from Siril.

Type renvoyé
str

Renvoie
Le répertoire de travail actuel sous forme de chaîne de caractères.

Lève
— *DataError* -- si aucune réponse n'a été obtenue,
— *SirilError* -- pour toutes les autres erreurs.

image_lock()

A context manager that handles claiming and releasing the processing thread.

This method is designed to be used with a *with* statement to ensure that the thread is properly claimed before processing and released after processing, even if an exception occurs during processing. It is preferable to use this context manager rather than manually calling claim_thread() and release_thread() as the context manager will ensure correct cleanup if an exception occurs.

Note that the image_lock() context should only be entered when the script itself is operating on the Siril image data. If the script is calling a Siril command to alter the Siril image then the context **must not** be entered or the Siril command will be unable to acquire the processing thread and will fail.

Exemple :

```
try:
    with siril.image_lock():
        # Get image data
        image_data = self.get_image_pixeldata()
        # Process image data
        processed_data = some_processing_function(image_data)
        # Set the processed image data
        siril.set_image_pixeldata(processed_data)
except ProcessingThreadBusyError:
    # Handle busy thread case
    pass
except ImageDialogOpenError:
    # Handle open dialog case
    pass
```

Lève

- ***ProcessingThreadBusyError*** -- If the thread is already in use
- ***ImageDialogOpenError*** -- Si une boîte de dialogue de traitement d'image est ouverte
- ***SirilError*** -- If the thread cannot be claimed or released for other reasons

info_messagebox(*my_string*, *modal=False*)

Send an information message to Siril. The maximum message length is 1022 bytes : longer messages will be truncated. This is intended for displaying informational messages more prominently than using the Siril log.

Paramètres

- **my_string** (str) -- Le message à afficher dans la boîte de dialogue d'information
- **modal** (Optional[bool]) -- Sets whether or not the message box should be modal and wait for completion or non-modal and allow the script to continue execution. Note that although a modal message box will block execution of the script, if a TKinter main loop is running events will continue to queue up, so if the message box is triggered by clicking a button then the user may click it while the message box is shown and trigger a second message box which will display immediately the first one is closed.

Renvoie

True if the info was successfully displayed, False otherwise

Type renvoyé

bool

Lève

SirilError -- si une erreur est survenue.

is_cli()

Check if the current instance is running in CLI mode. This method is useful to detect how the script was invoked and whether to show or not a GUI. This is False when the script is called by clicking in the Script menu, True otherwise.

Renvoie
True if running in CLI mode, False otherwise.

Type renvoyé
bool

`is_image_loaded`()

Vérifie si une seule image est chargée dans Siril.

Renvoie
True if a single image is loaded, False if a single image is not loaded

Type renvoyé
bool

Lève
— *DataError* -- si aucune réponse n'est reçue,
— *SirilError* -- pour toutes les autres erreurs.

`is_sequence_loaded`()

Vérifie si une séquence est chargé dans Siril.

Renvoie
True if a sequence is loaded, False if a sequence is not loaded

Type renvoyé
bool

Lève
— *DataError* -- si aucune réponse n'est reçue,
— *SirilError* -- pour toutes les autres erreurs.

`load_image_from_file`(*filepath*, *with_pixels=True*, *preview=False*, *linked=False*)

Request Siril to load an image from a file and transfer it to sirilpy. This method does not change the image currently loaded in Siril. Any image format supported by Siril is supported. This may be used as an alternative to loading an image using astropy.io.fits, however perhaps the main benefit to using it is that it supports the preview option which can be used to obtain an 8-bit autostretched rendering of the image more quickly than is possible using astropy and applying an autostretch using numpy.

Paramètres
— **filepath** (`str`) -- Chaîne de caractères indiquant le chemin d'accès au fichier image à charger.
— **with_pixels** (`Optional[bool]`) -- bool specifying whether or not to return the pixel data for the image (default is True).
— **preview** (`Optional[bool]`) -- bool specifying whether or not to return the real pixel data or an autostretched uint8_t preview version. Only has an effect in conjunction with with_pixels = True
— **linked** (`Optional[bool]`) -- bool specifying whether the autostretch preview should be linked or not. Has no effect unless preview is True.

Type renvoyé
`Optional[FFit]`

Renvoie
FFit object containing the image data

Lève
— **FileNotFoundError** -- si le fichier indiqué n'existe pas,
— *DataError* -- en cas de réception de données erronées,
— *SirilError* -- si une erreur est survenue.

`log`(*my_string*, *color=LogColor.DEFAULT*)

Send a log message to Siril. The maximum message length is 1022 bytes : longer messages will be truncated.

Paramètres

— **my_string** (str) -- Le message à loger
— **color** (*LogColor*) -- Defines the text color, defaults to white. See the documentation
— **which** (*for LogColor for an explanation of which colors should be used for*)
— **purposes.**

Lève
SirilError -- si la commande échoue

Type renvoyé
bool

open_dialog(*dialog*)

Opens a Siril GUI dialog. Introduced in sirilpy version 1.0.20.

Paramètres
dialog (DialogID) -- Specifies the dialog to open.

Renvoie
True if the dialog-opening action succeeded False if the dialog-opening action failed (e.g. because the criteria were not met)

Lève
— **TypeError** -- if the parameter is not a DialogID.
— *SirilError* -- if the method is called headless or an error occurs.

overlay_add_polygon(*polygon*)

Adds a user polygon to the Siril display overlay :type polygon : *Polygon* :param polygon : Polygon defining the polygon to be added

Renvoie
les entrées mise à jours avec l'ID assigné par Siril

Type renvoyé
Polygon

Lève
— **TypeError** -- données non valides fournies,
— *SirilConnectionError* -- en cas d'échec de la connexion,
— *DataError* -- en cas de réception de données non valides,
— *SirilError* -- pour toutes défaillance

overlay_clear_polygons()

Supprime tous les polygones utilisateur de l'affichage Siril

Renvoie
True si la commande a abouti, False dans le cas contraire

Type renvoyé
bool

Lève
— *SirilError* -- si une erreur s'est produite
— *SharedMemoryError* -- si une erreur de mémoire partagée s'est produite

overlay_delete_polygon(*polygon_id*)

Deletes a single user polygon from the Siril overlay, specified by ID

Paramètres
id -- int specifying the polygon ID to be deleted

Lève
SirilError -- en cas d'échec

`overlay_draw_polygon`(*color=16711744*, *fill=False*)

Enters a mode where the user can draw a Polygon in the Siril window by clicking the main mouse button and dragging. Releasing the mouse button finalises and closes the polygon.

Paramètres
- **color** -- uint32 specifying packed RGBA values. Default : 0x00FF0040, 75% transparent green)
- **fill** -- bool specifying whether or not to fill the polygon (default : False)

`overlay_get_polygon`(*polygon_id*)

Gets a single user polygon from the Siril overlay, specified by ID

Paramètres
id -- int specifying the polygon ID to be retrieved. The special ID -1 will

retrieve the most recently added polygon.

Renvoie
le polygone spécifié s'il existe, None sinon

Type renvoyé
Polygon

Lève
SirilError -- en cas d'échec

`overlay_get_polygons_list`()

Gets a List of all user polygons from the Siril overlay

Renvoie
the list of Polygon if some exist, None otherwise

Type renvoyé
List[*Polygon*]

Lève
- ***SirilError*** -- si une erreur s'est produite
- ***SharedMemoryError*** -- si une erreur de mémoire partagée s'est produite

`pix2radec`(*x*, *y*)

Converts a pair of pixel coordinates into RA and dec coordinates using the WCS of the image loaded in Siril. This requires that an image is loaded in Siril and that it has been platesolved (i.e. it has a WCS solution).

Paramètres
- **x** (`float`) -- float : provides the x coordinate to be converted
- **y** (`float`) -- float : provides the y coordinate to be converted

Renvoie
(RA, Dec) sous la forme d'un tuple de deux floats.

Type renvoyé
Tuple[float, float]

Lève
- ***NoImageError*** -- Si aucune image ou séquence n'est chargé,
- **ValueError** -- If the image or loaded sequence frame is not plate solved,
- ***SirilError*** -- En cas d'erreurs survenant lors de l'exécution de pix2radec.

`radec2pix`(*ra*, *dec*)

Converts a pair of RA,dec coordinates into image pixel coordinates using the WCS of the image loaded in Siril. This requires that an image is loaded in Siril and that it has been platesolved (i.e. it has a WCS solution).

Paramètres
- **ra** (`float`) -- float : provides the RA coordinate to be converted
- **dec** (`float`) -- float : provides the dec coordinate to be converted

Renvoie
[x,y] comme un Tuple de deux nombres flottants.

Type renvoyé
Tuple[float, float]

Lève
— *NoImageError* -- Si aucune image ou séquence n'est chargé,
— **ValueError** -- If the image or loaded sequence frame is not plate solved,
— *SirilError* -- En cas d'erreurs survenant lors de l'exécution de radec2pix.

redo()

Redoes the last undone operation, if there is an undo history and an undone operation available to be redone.

reset_progress()

Réinitialise la barre de progression de Siril.

Paramètres
none

Lève
SirilError -- Pour toutes erreurs.

Type renvoyé
bool

save_image_file(*data*, *header=None*, *filename=None*)

Save image pixeldata and metadata to a FITS file. This uses Siril to save the image and can therefore be used to avoid a script dependency on astropy if it is only required for saving an image. This allows images to be saved directly from python without having to use set_image_pixeldata() and set_image_metadata_from_header_string() and saving the Siril image. It support processing multiple images without affecting the image currently loaded in Siril.

Paramètres
— **data** -- Either a numpy.ndarray containing the image data (must be 2D or 3D array with dtype float32 or uint16), OR a FFit object containing both data and header.
— **header** -- string containing the FITS header data. Required if data is a numpy array, ignored if data is a FFit object.
— **filename** -- string containing the path where the file should be saved. Required if data is a numpy array. If data is a FFit object and filename is None, will use fit.filename.

Renvoie
True en cas de succès, False dans le cas contraire

Type renvoyé
bool

Lève
— **ValueError** -- if the input array or header is invalid,
— **TypeError** -- if invalid parameter types are provided,
— *SirilError* -- si une erreur s'est produite lors du traitement de la commande.

Exemples

Using numpy array and header string siril.save_image_file(data_array, header_string, "output.fit")

Using FFit object siril.save_image_file(fit, filename="output.fit")

Using FFit object with its own filename siril.save_image_file(fit)

set_image_bgsamples(*points*, *show_samples=False*, *recalculate=True*)

Serialize a set of background sample points and send via shared memory. Points can be provided either as : - List of (x,y) Tuples : BGSamples are created with these positions and Siril will automatically compute the statistics. - List of BGSample objects : The complete sample data is sent to Siril. By default Siril will recalculate statistics for the sample points on receipt, but this can be overridden with the argument recalculate=False

Paramètres

- **points** (Union[List[Tuple[float, float]], List[*BGSample*]]) -- List of sample points, either as (x,y) tuples or BGSample objects
- **show_samples** (bool) -- Afficher ou non les points d'échantillonnage dans Siril
- **recalculate** -- Whether to recalculate the sample points once set. This only applies if the sample points are provided as a List of BGSamples, in which case it defaults to True. If the sample points are provided as a List of (x,y) Tuples then the parameter has no effect. Setting recalculate=False is usually a bad idea but the option is provided to support potential advanced uses where the values are adjusted in python code to manipulate the background fit.

Retourne : True si la commande a abouti, False dans le cas contraire

Lève

- *NoImageError* -- si aucune image n'est chargée dans Siril,
- **ValueError** -- if samples do not have valid positions,
- *SirilError* -- if there was a Siril error in handling the command.

set_image_filename(*filename*)

Set the image filename in Siril. :type filename : str :param filename : the image filename to set :type filename : str

Lève

- *SirilError* -- si une erreur est survenue.
- **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie

True si le nom de fichier a été défini avec succès

Type renvoyé

bool

set_image_iccprofile(*iccprofile*)

Configurer le profil ICC de l'image dans Siril

Paramètres

- **iccprofile** (*bytes*) -- The ICC profile to send to Siril. This will
- **profile** (*replace an existing ICC*)
- **None** (*if one is set. If*)

:param : :param any existing image ICC profile will be removed. :

Retourne : True si la commande a abouti, False dans le cas contraire

Lève

- *NoImageError* -- si aucune image n'est chargée dans Siril,
- **ValueError** -- if iccprofile is not valid type,
- *SirilError* -- if there was a Siril error in handling the command.

set_image_metadata_from_header_string(*header*)

Send image metadata to Siril from a FITS header. The header can be obtained from a sirilpy FFit.header or alternatively from a FITS file obened from disk using astropy.fits.

Exemple :

```
hdul = fits.open('your_fits_file.fits')
# Get the header from the primary HDU (or any other HDU you want)
header = hdul[0].header
# Convert the header to string
header_string = header.tostring(sep='\\n')
# Send the metadata to Siril
siril.set_image_metadata_from_header_string(header_string)
```

Paramètres
header (str) -- string containing the FITS header data

Renvoie
True en cas de succès, False dans le cas contraire

Type renvoyé
bool

Lève
— **TypeError** -- paramètre invalide fourni,
— *NoImageError* -- si aucune image n'est chargée dans Siril,
— *SirilError* -- si une erreur est survenue.

`set_image_pixeldata`(*image_data*)

Envoyer les données d'image à Siril à l'aide de la mémoire partagée.

Paramètres
image_data (ndarray) -- numpy.ndarray containing the image data. Must be 2D (single channel) or 3D (multi-channel) array with dtype either np.float32 or np.uint16.

Lève
— *NoImageError* -- si aucune image n'est chargée dans Siril,
— **ValueError** -- Si le tableau d'entré est invalide,
— *SirilError* -- si une erreur s'est produite lors du traitement de la commande.

Type renvoyé
bool

`set_seq_frame_incl`(*index*, *incl*)

Set whether given frame(s) are included in the currently loaded sequence in Siril. This method is intended for use in creating custom sequence filters.

Paramètres
— **index** (Union[int, List[int]]) -- integer or list of integers specifying which frame(s) to set the inclusion status for. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame number 1. Passing a list is available since sirilpy 1.0.17
— **incl** (bool) -- bool specifying whether the frame(s) are included or not.

Lève
— *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
— **TypeError** -- si l'index n'est pas un entier ou une liste d'entiers,
— *SirilError* -- en cas d'échec.

`set_seq_frame_pixeldata`(*index*, *image_data*, *prefix*)

Send sequence frame image data to Siril using shared memory. Note that this method only works with sequences of FITS images : it does **not** work with FITSEQ, SER or AVI single-file sequences. The image_lock() context manager is not required in order to use this method.

Paramètres
— **index** (int) -- integer specifying which frame to set the pixeldata for. This uses a 0-based indexing scheme, i.e. the first frame is frame number 0, not frame numer 1.
— **image_data** (ndarray) -- numpy.ndarray containing the image data. Must be 2D (single channel) or 3D (multi-channel) array with dtype either np.float32 or np.uint16.
— **prefix** (str) -- String prefix to use when saving the file to make a new sequence. Note that saving sequence frames with a new prefix does not by itself create a new sequence : once all the frames have been saved with the new sequence prefix, `sirilpy.SirilInterface.create_new_seq()` must be called to create the actual sequence file. Note that while it is permitted to pass prefix=None, this will overwrite the existing sequence and is not typically what is wanted, therefore the parameter is not optional and must be passed explicitly.

Lève

— *NoSequenceError* -- si aucune séquence n'est chargée dans Siril,
— **ValueError** -- Si le tableau d'entré est invalide,
— *SirilError* -- si une erreur s'est produite lors du traitement de la commande.

Type renvoyé
bool

set_siril_pan(*xoff*, *yoff*)

Set the display offset (pan) in Siril to (xoff, yoff). :type xoff : float :param xoff : x display offset :type xoff : float :type yoff : float :param yoff : y display offset :type yoff : float

Lève
— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si le décalage d'affichage a été défini avec succès

Type renvoyé
bool

set_siril_selection(*x=None*, *y=None*, *w=None*, *h=None*, *selection=None*)

Défini la sélection d'image dans Siril à l'aide des coordonnées et des dimensions fournies.

Paramètres
— **x** (Optional[int]) -- X-coordinate of the selection's top-left corner (must be provided with y, w, h)
— **y** (Optional[int]) -- Y-coordinate of the selection's top-left corner (must be provided with x, w, h)
— **w** (Optional[int]) -- Width of the selection (must be provided with x, y, h)
— **h** (Optional[int]) -- Height of the selection (must be provided with x, y, w)
— **selection** (Optional[Tuple[int, int, int, int]]) -- A tuple of (x, y, w, h) as returned by get_siril_selection()

Lève
— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si la sélection a été effectuée avec succès

Type renvoyé
bool

set_siril_slider_lohi(*lo=None*, *hi=None*)

Set the slider values in Siril using the provided lo and hi values. If the sliders mode is not already set to USER, it is set to that mode as setting slider values is only relevant in that mode. :type lo : Union[float, int] :param lo : Low value for the slider (float [0,1] or uint16) :type hi : Union[float, int] :param hi : High value for the slider (float [0,1] or uint16)

Lève
— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si l'état du curseur a été défini correctement

Type renvoyé
bool

set_siril_slider_mode(*mode*)

Set the slider state in Siril using the provided lo, hi and mode values. :type mode : *SlidersMode* :param mode : SlidersMode enum value

Lève

— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si l'état du curseur a été défini correctement

Type renvoyé
bool

set_siril_stf(*mode*)

Set the screen transfer function in Siril using the provided enum value. :type mode : *SlidersMode* :param mode : STFType enum value

Lève
— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si l'état du curseur a été défini correctement

Type renvoyé
bool

set_siril_stf_linked(*state*)

Set the screen transfer function in Siril using the provided enum value. :type state : bool :param state : bool

Lève
— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si l'état du curseur a été défini correctement

Type renvoyé
bool

set_siril_zoom(*zoom*)

Set the display zoom. Passing any negative value will set 'zoom to fit'. :type zoom : float :param zoom : zoom level :type zoom : float

Lève
— *SirilError* -- si une erreur est survenue.
— **ValueError** -- si les paramètres ne sont pas correctement fournis.

Renvoie
True si le décalage d'affichage a été défini avec succès

Type renvoyé
bool

undo()

Undoes the last operation, if there is an undo history available.

undo_save_state(*my_string*)

Saves an undo state. The maximum message length is 70 bytes : longer messages will be truncated. Requires a single image to be loaded.

Paramètres
my_string (str) -- Le message à écrire dans le FITS HISTORY

Renvoie
True if the message was successfully logged, False if a logging error occurred or None if the script is headless (undo does not happen when headless)

Type renvoyé
Optional[bool]

Lève
SirilError -- si une erreur est survenue.

update_progress(*message*, *progress*)

Send a progress update to Siril with a message and completion percentage.

Paramètres
- **message** (str) -- Message de statu à afficher
- **progress** (float) -- Progress value in the range 0.0 to 1.0. The following special values can be used : -1.0 will pulsate the progress bar, and -2.0 will update the progress bar text but will not update the progress shown in the bar.

Lève
- **ValueError** -- Si l'argument progress est hors limites,
- *SirilError* -- Pour tout autre erreurs.

Type renvoyé
bool

warning_messagebox(*my_string*, *modal=False*)

Send a warning message to Siril. The maximum message length is 1022 bytes : longer messages will be truncated. This is intended for displaying warning messages more prominently than using the Siril log.

Paramètres
- **my_string** (str) -- Le message à afficher dans la boite de message d'avertissement
- **modal** (Optional[bool]) -- Sets whether or not the message box should be modal and wait for completion or non-modal and allow the script to continue execution. Note that although a modal message box will block execution of the script, if a TKinter main loop is running events will continue to queue up, so if the message box is triggered by clicking a button then the user may click it while the message box is shown and trigger a second message box which will display immediately the first one is closed.

Renvoie
True si l'avertissement a correctement été affiché, False sinon

Type renvoyé
bool

Lève
SirilError -- si une erreur est survenue.

xy_plot(*plot_data*, *display=True*, *save=False*)

Serialize plot data and send via shared memory. See the sirilpy.plot submodule documentation for how to configure a PlotData object for use with SirilInterface.xy_plot()

Paramètres
- **plot_metadata** -- Objet PlotMetadata contenant la configuration du graphique
- **display** -- booléen indiquant s'il faut afficher le graphique à l'écran (valeur par défaut : True)
- **save** -- bool indicating whether to save to the file specified in PlotData.savename (defaults to False)

Lève
- *DataError* -- if invalid xy_plot data is received via shared memory,
- *SirilError* -- Si une erreur se produit.

25.4 Modèles de données Sirilpy

This submodule provides dataclasses to represent the main Siril data structures. Most dataclasses have corresponding deserialization methods that are used by the SirilInterface methods. All its members are made available at the module root level, there is no need to import models separately.

class sirilpy.models.**BGSample**(*x=None*, *y=None*, *position=None*, *size=25*, ***kwargs*)

Bases : `object`

Python equivalent of the Siril background_sample struct. Used to hold background sample data obtained from Siril, or to generate or modify background sample data to set in Siril. A BGSample can be constructed as : - s1 = BGSample(x=1.0, y=2.0) - s2 = BGSample(position=(1.0, 2.0)) - s3 = BGSample(x=1.0, y=2.0, mean=0.5, size=31)

classmethod deserialize(*data*)

Deserialize a portion of a buffer into a BGSample object

Paramètres
data (*bytes*) -- The full binary buffer containing BGSample data

Renvoie
Un objet BGSample

Type renvoyé
BGSample

Lève
— **ValueError** -- If the buffer slice size does not match the expected size.
— **struct.error** -- If there is an error unpacking the binary data.

max: float = 0.0

mean: float = 0.0

median: Tuple[float, float, float] = (0.0, 0.0, 0.0)

Valeur médiane des canaux R, V et B. Pour les images mono, seul median[0] est utilisé.

min: float = 0.0

position: Tuple[float, float] | None = None

Position (x, y) en coordonnées image

size: int = 25

The default size matches the size of Siril bg samples.

valid: bool = True

Par défaut, les échantillons sont considérés comme valides

class sirilpy.models.**DistoData**(*index=DistoType.DISTO_UNDEF*, *filename=''*, *velocity=(0, 0)*)

Bases : `object`

Python equivalent of Siril disto_params structure

filename: str = ''

filename if DISTO_FILE or DISTO_MASTER (and optional for DISTO_FILE_COMET)

index: *DistoType* = 0

Spécifie le type de distortion

velocity: Tuple[float, float] = (0, 0)

vitesse de décalage si DISTO_FILE_COMET

class sirilpy.models.**FFit**(*bitpix=None*, *orig_bitpix=None*, *naxis=0*, *_naxes=(0, 0, 0)*, *keywords=<factory>*, *checksum=False*, *header=None*, *unknown_keys=None*, *stats=<factory>*, *mini=0.0*, *maxi=0.0*, *neg_ratio=0.0*, *_data=None*, *top_down=False*, *_focalkey=False*, *_pixelkey=False*, *history=<factory>*, *color_managed=False*, *_icc_profile=None*)

Bases : `object`

Python equivalent of Siril ffit (FITS) structure, holding image pixel data and metadata.

allocate_data()

Allocate memory for image data with appropriate type. self.width, self.height, self.naxis, self.naxes and self.dtype must be set before calling this method.

Lève
ValueError -- if self.bitpix is not set to BitpixType.USHORT_IMG or BitpixType.FLOAT_IMG

ensure_data_type(*target_type=None*)

Ensure data is in the correct type with proper scaling

Paramètres
target_type -- Optional np.dtype to convert to. If None, uses self.type

Lève
ValueError -- if the conversion is between data types that are not internally used by Siril for calculation

estimate_noise(*array*, *nullcheck=True*, *nullvalue=0.0*)

Estimate the background noise in the input image using the sigma of first-order differences.

noise = 1.0 / sqrt(2) * RMS of (flux[i] - flux[i-1])

Paramètres
— **array** (*np.ndarray*) -- Tableau 2D de pixels d'image (np.uint16 ou np.float32).
— **nullcheck** (*bool*) -- Si la valeur est True, vérifiez s'il y a des valeurs nulles.
— **nullvalue** (Optional[float]) -- The value of null pixels (only used if nullcheck is True).

Renvoie
Valeur estimé du bruit.

Type renvoyé
float

Lève
ValueError -- si le tableau n'a pas la bonne taille

get_channel(*channel*)

Get a specific channel of the pixel data. Note that this does not pull pixel data directly from the image loaded in Siril : that must previously have been obtained using get_image_pixeldata() or get_image()

Type renvoyé
ndarray

update_stats()

Update image statistics for all channels. Note that this only updates the statistics based on the NumPy array representing pixel data in the python FFit object, it does not update the statistics of the image in Siril.

bitpix: *BitpixType* | None = None

FITS header specification of the image data type.

property channels: int

Canaux d'image

checksum: bool = False

Whether Siril will write FITS data checksums for this file.

color_managed: bool = False

Specifies whether the image is color managed or not.

property data: ndarray | None

The pixel data of the current image loaded in Siril, stored as a NumPy array

header: str | None = None

L'entête FITS en temps que chaîne de caractère.

property height: int

Hauteur de l'image

history: list[str]

Contains a list of strings holding the HISTORY entries for this image.

property icc_profile: bytes | None

The ICC profile as raw bytes data. This may be converted for use by modules such as pillow which can handle ICC profiles.

keywords: *FKeywords*

A FKeywords object containing FITS header keywords.

maxi: float = 0.0

La valeur maximal à travers tous les canaux de l'image.

mini: float = 0.0

La valeur minimal à travers tous les canaux de l'image.

property naxes: Tuple[int, int, int]

The naxes tuple holds the image dimensions as width x height x channels. Note that the axis ordering differs between Siril representation as held in naxes and numpy representation as held in _data.shape (which is channels x height x width)

naxis: int = 0

Le nombre d'axes (2 pour une image mono, 3 pour une image RGB). Correspond au mot clé FITS NAXIS.

neg_ratio: float32 = 0.0

Le rapport entre le nombre de pixels négatifs et le nombre total de pixels.

orig_bitpix: *BitpixType* | None = None

FITS header specification of the original image data type.

stats: List[*ImageStats* | None]

A list of ImageStats objects, one for each channel.

top_down: bool = False

Specifies the ROWORDER for this image. The FITS specification directs that FITS should be stored bottom-up, but many CMOS sensors are natively TOP_DOWN and capture software tends to save FITS images captured by these sensors as TOP_DOWN.

unknown_keys: str | None = None

All unknown FITS header keys as a string. This gives access to header cards that Siril does not use internally.

property width: int

Largeur de l'image

class sirilpy.models.**FKeywords**(*bscale=1.0*, *bzero=0.0*, *lo=0*, *hi=0*, *flo=0.0*, *fhi=0.0*, *program=''*, *filename=''*, *row_order=''*, *filter=''*, *image_type=''*, *object=''*, *instrume=''*, *telescop=''*, *observer=''*, *bayer_pattern=''*, *sitelat_str=''*, *sitelong_str=''*, *focname=''*, *date=None*, *date_obs=None*, *data_max=0.0*, *data_min=0.0*, *pixel_size_x=0.0*, *pixel_size_y=0.0*, *binning_x=1*, *binning_y=1*, *expstart=0.0*, *expend=0.0*, *bayer_xoffset=0*, *bayer_yoffset=0*, *airmass=1.0*, *focal_length=0.0*, *flength=0.0*, *iso_speed=0.0*, *exposure=0.0*, *aperture=0.0*, *ccd_temp=0.0*, *set_temp=0.0*, *livetime=0.0*, *stackcnt=0*, *cvf=0.0*, *gain=0*, *offset=0*, *focuspos=0*, *focussz=0*, *foctemp=0.0*, *centalt=0.0*, *centaz=0.0*, *sitelat=0.0*, *sitelong=0.0*, *siteelev=0.0*, *objctra=None*, *objctdec=None*, *ra=None*, *dec=None*, *pltsolvd=False*, *pltsolvd_comment=None*)

Bases : *object*

Python equivalent of Siril fkeywords structure. Contains the FITS header keyword values converted to suitable data types.

classmethod deserialize(*data*)

Deserialize binary keyword-block (the block produced by keywords_to_py()) into an FKeywords object. Raises ValueError on size mismatch and SirilError for other unpacking issues.

Type renvoyé
FKeywords

FLEN_VALUE: ClassVar[int] = 71

KEYWORDS_FORMAT: ClassVar[str] = '!
71s71s71s71s71s71s71s71s71s71s71s71s71s71s71sddQQddddddQQddddddddqqddddddddddQdqqqqdqqdd?
'

KEYWORDS_SIZE: ClassVar[int] = 1465

airmass: float = 1.0

Masse d'air au centre du cadre (Gueymard 1993)

aperture: float = 0.0

Valeur de l'ouverture en nombre flottant

bayer_pattern: str = ''

Matrice de Bayer

bayer_xoffset: int = 0

X offset of the Bayer pattern

bayer_yoffset: int = 0

Y offset of the Bayer pattern

binning_x: int = 1

XBINNING FITS header card as an int

binning_y: int = 1

mot clé YBINNING de l'en-tête FITS en int

bscale: float = 1.0

Décalage de la plage de données par rapport à celle d'une valeur non signée (unsigned short)

bzero: float = 0.0

Facteur d'échelle par défaut

ccd_temp: float = 0.0

Température du CCD en nombre à virgule flotante

centalt: float = 0.0

[deg] Altitude du télescope

centaz: float = 0.0

[deg] Azimut du télescope

cvf: float = 0.0

Facteur de conversion (e-/ ADU)

data_max: float = 0.0

used to check if 32b float is in the [0, 1] range

data_min: float = 0.0

used to check if 32b float is in the [0, 1] range

date: datetime | None = None

Date UTC de création du fichier FITS

date_obs: datetime | None = None

ss début de l'observation, UT

Type
YYYY-MM-DDThh

Type
mm

dec: float | None = None
Dec en nombre flottant, si disponible
expend: float = 0.0
Fin de l'exposition sous forme de date julienne
exposure: float = 0.0
Temps d'exposition en nombre flottant (s)
expstart: float = 0.0
Début de l'exposition sous forme de date julienne
fhi: float32 = 0.0
MIPS-Hi key in FITS file, "Upper visualization cutoff (float)"
filename: str = ''
Nom de fichier d'origine
filter: str = ''
Nom du filtre actif
flength: float = 0.0
[mm] Longueur focale
flo: float32 = 0.0
MIPS-LO key in FITS file, "Lower visualization cutoff (float)"
focal_length: float = 0.0
[mm] Longueur focale
focname: str = ''
Nom de l'équipement de mise au point
foctemp: float = 0.0
Température du focuseur
focuspos: int = 0
Position du focuseur
focussz: int = 0
[um] Taille du pas du focuseur
gain: int = 0
Facteur de gain lu depuis la caméra
hi: int = 0
MIPS-HI key in FITS file, "Upper visualization cutoff"
image_type: str = ''
Types d'image
instrume: str = ''
Nom de l'instrument
iso_speed: float = 0.0
Valeur de sensibilité ISO sous forme de nombre à virgule flottante
livetime: float = 0.0
Somme des temps d'expositions (s)
lo: int = 0
MIPS-LO key in FITS file, "Lower visualization cutoff"
objctdec: str | None = None
objet Dec sous forme de string, si disponible
objctra: str | None = None
objet Ra sous forme de string, si disponible

object: str = ''
Nom de l'objet d'intérêt
observer: str = ''
Nom de l'observateur
offset: int = 0
Valeur d'offset lue dans la caméra
pixel_size_x: float = 0.0
mot clé XPIXSZ de l'entête FITS en float
pixel_size_y: float = 0.0
mot clé YPIXSZ de l'entête FITS en float
pltsolvd: bool = False
si résolution astrométrique ou non
pltsolvd_comment: str | None = None
commentaire sur a solution astrométrique
program: str = ''
Logiciel qui a créé ce HDU
ra: float | None = None
Ra en nombre flottant, si disponible
row_order: str = ''
Ordre des lignes dans le tableau d'images
set_temp: float = 0.0
CCD set temperature as a float
siteelev: float = 0.0
[m] Élévation du site d'observation
sitelat: float = 0.0
sitelat_str: str = ''
sitelong: float = 0.0
[deg] Longitude du site d'observation
sitelong_str: str = ''
stackcnt: int = 0
Nombres d'images empilé
telescop: str = ''
Télescope utilisé pour acquérir cette image

class sirilpy.models.**FPoint**(*x*, *y*)

Bases : `object`

Represents a 2D point with float x and y coordinate values in the Siril image.

x: float
coordonnée x
y: float
coordonnée y

class sirilpy.models.**Homography**(*h00=0.0*, *h01=0.0*, *h02=0.0*, *h10=0.0*, *h11=0.0*, *h12=0.0*, *h20=0.0*, *h21=0.0*, *h22=0.0*, *pair_matched=0*, *Inliers=0*)

Bases : `object`

Python equivalent of the Siril Homography structure. Contains coefficients for the Homography matrix that maps a sequence frame onto the reference frame.

Inliers: int = 0

number of inliers kept after RANSAC step

h00: float = 0.0

Matrice d'homographie H00

h01: float = 0.0

Matrice d'homographie H01

h02: float = 0.0

Matrice d'homographie H02

h10: float = 0.0

Matrice d'homographie H10

h11: float = 0.0

Matrice d'homographie H11

h12: float = 0.0

Matrice d'homographie H12

h20: float = 0.0

Matrice d'homographie H20

h21: float = 0.0

Matrice d'homographie H21

h22: float = 0.0

Matrice d'homographie H22

pair_matched: int = 0

nombre de paires trouvées

class sirilpy.models.**ImageAnalysis**(*bgnoise=0.0*, *fwhm=0.0*, *wfwhm=0.0*, *nbstars=0*, *roundness=0.0*, *imagetype=0*, *timestamp=0*, *channels=0*, *height=0*, *width=0*, *filter=''*)

Bases : `object`

Structure to hold image analysis data, used for culling

classmethod deserialize(*data*)

Unpack a network-safe binary struct into an ImageAnalysis instance.

Type renvoyé

ImageAnalysis

serialize()

Pack the dataclass into a network-safe binary struct.

Type renvoyé

bytes

FLEN: ClassVar[int] = 71

bgnoise: float = 0.0

Bruit de fond RMS

channels: int = 0

number of channels in the image

filter: str = ''

filter name (fixed-length string from C, max 70 chars)

fwhm: float = 0.0

Moyenne fwhm

height: int = 0

hauteur de l'image

imagetype: *ImageType* = 0

Image type enum

nbstars: int = 0

Nombre d'étoiles détectées

roundness: float = 0.0

Moyenne de la rondeur des étoiles

timestamp: int = 0

00 UTC)

Type

UNIX timestamp (64-bit seconds since 1970/1/1 00

wfwhm: float = 0.0

FWHM pondérée moyenne

width: int = 0

largeur de l'image

class sirilpy.models.**ImageStats**(*total=0*, *ngoodpix=0*, *mean=0.0*, *median=0.0*, *sigma=0.0*, *avgDev=0.0*, *mad=0.0*, *sqrtbwmv=0.0*, *location=0.0*, *scale=0.0*, *min=0.0*, *max=0.0*, *normValue=0.0*, *bgnoise=0.0*)

Bases : `object`

Python equivalent of Siril imstats structure. Contains statistics for a particular channel of a Siril image.

classmethod deserialize(*data*)

Deserialize binary data into an ImageStats object.

Paramètres

data (`bytes`) -- (bytes) Binary data to unpack

Type renvoyé

ImageStats

Renvoie : un objet ImageStats

Raises : SirilError : Si la taille des données reçues est incorrecte

struct.error : En cas d'échec du décompressage

avgDev: float = 0.0

écart moyen des pixels

bgnoise: float = 0.0

Bruit de fond RMS

location: float = 0.0

position des valeurs de pixels

mad: float = 0.0

écart moyen des pixels

max: float = 0.0

valeur maximale du pixel

mean: float = 0.0

valeur moyenne des pixels

median: float = 0.0

valeur médiane des pixels

min: float = 0.0

valeur minimale du pixel

ngoodpix: int = 0

nombre de pixels différents de zéro

normValue: float = 0.0

valeur normé des pixels

scale: float = 0.0

Valeur de normalisation des pixels

sigma: float = 0.0

déviation standard des pixels

sqrtbwmv: float = 0.0

square root of the biweight midvariance of pixel values

total: int = 0

nombre total de pixels

class sirilpy.models.**ImgData**(*filenum=0*, *incl=False*, *date_obs=None*, *airmass=0.0*, *rx=0*, *ry=0*)

Bases : `object`

Python equivalent of Siril imgdata structure

classmethod deserialize(*response*)

Deserialize binary response into an ImgData object.

Paramètres

response (*bytes*) -- Données binaires à décompresser.

Renvoie

An ImgData object with deserialized data.

Type renvoyé

ImgData

Lève

— **ValueError** -- Si la taille des données reçues est incorrecte.

— **struct.error** -- Si la décompression échoue.

airmass: float = 0.0

airmass de l'image

date_obs: datetime | None = None

date de l'observation

filenum: int = 0

real file index in the sequence

incl: bool = False

sélectionné dans la séquence

rx: int = 0

width

ry: int = 0

height

class sirilpy.models.**PSFStar**(*star_name=None*, *B=0.0*, *A=0.0*, *x0=0.0*, *y0=0.0*, *sx=0.0*, *sy=0.0*, *fwhmx=0.0*, *fwhmy=0.0*, *fwhmx_arcsec=0.0*, *fwhmy_arcsec=0.0*, *angle=0.0*, *rmse=0.0*, *sat=0.0*, *R=0*, *has_saturated=False*, *beta=0.0*, *profile=StarProfile.GAUSSIAN*, *xpos=0.0*, *ypos=0.0*, *mag=0.0*, *Bmag=0.0*, *s_mag=999.99*, *s_Bmag=999.99*, *SNR=0.0*, *phot_is_valid=False*, *BV=0.0*, *B_err=0.0*, *A_err=0.0*, *x_err=0.0*, *y_err=0.0*, *sx_err=0.0*, *sy_err=0.0*, *ang_err=0.0*, *beta_err=0.0*, *layer=0*, *units=None*, *ra=0.0*, *dec=0.0*)

Bases : `object`

Python equivalent of the Siril fwhm_struct structure. Contains data on a modelled fit to a star identified in the image.

classmethod deserialize(*data*)

Deserialize a portion of a buffer into a PSFStar object.

Paramètres
data (bytes) -- (bytes) The full binary buffer containing PSFStar data.

Type renvoyé
PSFStar

Renvoie
Objet PSFStar

Lève
— **ValueError** -- If the buffer slice size does not match the expected size.
— **struct.error** -- If there is an error unpacking the binary data.

A: float = 0.0

amplitude

A_err: float = 0.0

erreur en A

B: float = 0.0

valeur moyenne du fond du ciel

BV: float = 0.0

only used to pass data in photometric color calibration

B_err: float = 0.0

erreur en B

Bmag: float = 0.0

magnitude B

R: int = 0

Optimized box size to enclose sufficient pixels in the background

SNR: float = 0.0

SNR de l'étoile

ang_err: float = 0.0

erreur sur l'angle

angle: float = 0.0

angle of the x and yaxes with respect to the image x and y axes

beta: float = 0.0

Paramètre beta de l'équation Moffat

beta_err: float = 0.0

erreur en beta

dec: float = 0.0

Déclinaison

fwhmx: float = 0.0

FWHM selon l'axe x en pixels

fwhmx_arcsec: float = 0.0

FWHM sur l'axe x en arc seconds

fwhmy: float = 0.0

FWHM selon l'axe y en pixels

fwhmy_arcsec: float = 0.0

FWHM sur l'axe xyen arc seconds

has_saturated: bool = False

Shows whether the star is saturated or not

layer: int = 0

image channel on which the star modelling was carried out

mag: float = 0.0

(V) magnitude, approximatif ou précis

phot_is_valid: bool = False

True if mag, Bmag, s_mag, s_Bmag and SNR are photometrically accurate with no saturated pixels. Introduced in sirilpy 1.0.4.

profile: *StarProfile* = 0

ra: float = 0.0

Ascension droite

rmse: float = 0.0

RMSE of the minimization

s_Bmag: float = 999.99

Erreur sur la magnitude B

s_mag: float = 999.99

erreur sur la magnitude (V)

sat: float = 0.0

Niveau au dessus duquel les pixels sont saturés

star_name: str | None = None

sx: float = 0.0

Size of the fitted function on the x axis in PSF coordinates

sx_err: float = 0.0

erreur en sx

sy: float = 0.0

Size of the fitted function on the y axis in PSF coordinates

sy_err: float = 0.0

erreur en sy

units: str | None = None

Unité

x0: float = 0.0

coordonnée x du pic

x_err: float = 0.0

erreur en x

xpos: float = 0.0

x position of the star in the image

y0: float = 0.0

coordonnée y du pic

y_err: float = 0.0

erreur en y

ypos: float = 0.0

y position of the star in the image

class sirilpy.models.**Polygon**(*points*, *polygon_id=0*, *color=4294967295*, *fill=False*, *legend=None*)

Bases : object

Represents a user-defined polygon. These can be filled or outline-only, and can have any color and transparency (alpha) value. They can also have an optional label which is displayed centred on the polygon.

Note that Polygons should be considered transitory if used with the overlay - they can be used to display information to the user but they may be cleared at any time if the user toggles the overlay button in the main Siril interface to clear the overlay.

classmethod deserialize_polygon(*data*)

Creates a Polygon object by deserializing a byte array.

Renvoie

A Polygon object and any remaining bytes in the byte
array. (The remaining bytes are for use in deserialize_polygon_list and can be safely ignored if deserializing a single polygon.)

Type renvoyé
Tuple

Lève
ValueError -- Si les données disponibles sont insuffisantes pour la désérialisation.

classmethod deserialize_polygon_list(*data*)

Creates a List of Polygon objects by deserializing a byte array.

Renvoie
Une liste d'objets Polygon.

Type renvoyé
List

Lève
ValueError -- If there is invalid data to deserialize.

classmethod from_rectangle(*rect*, ***kwargs*)

Create a Polygon from a rectangle of the kind returned by sirilpy.connection.get_siril_selection().

Paramètres
— **rect** (*Tuple[int, int, int, int]*) -- Rectangle tel que (x,y largeur, hauteur)
— ****kwargs** -- Additional keyword arguments to pass to Polygon constructor (polygon_id, color, fill, legend)

Renvoie
A new Polygon instance representing the rectangle

Type renvoyé
Polygon

contains_point(*x*, *y*)

Determine if a point is inside the polygon using Dan Sunday's optimized winding number algorithm.

This algorithm is more robust than ray casting for complex polygons and handles edge cases better, including points on edges and self-intersecting polygons.

Paramètres
— **x** (*float*) -- Coordonnée X du point à tester.
— **y** (*float*) -- Coordonnée Y du point à tester.

Renvoie
True if the point is inside the polygon, False otherwise.

Type renvoyé
bool

get_bounds()

Get the bounding box of the polygon.

Renvoie
(min_x, min_y, max_x, max_y)

Type renvoyé
Tuple[float, float, float, float]

Lève
ValueError -- Si le polygone n'a aucun point.

get_max_x()

Get the maximum x coordinate of the polygon.

Type renvoyé
float

get_max_y()

Get the maximum y coordinate of the polygon.

Type renvoyé
float

get_min_x()

Get the minimum x coordinate of the polygon.

Type renvoyé
float

get_min_y()

Get the minimum y coordinate of the polygon.

Type renvoyé
float

serialize()

Serializes a single Polygon object into a byte array.

Renvoie
A byte array representing the serialized polygon data.

Type renvoyé
bytes

Lève
ValueError -- Si le nombre de points dépasse la limite autorisée.

color: int = 4294967295

32-bit RGBA color (packed, uint_8 per component. Default value is 0xFFFFFFFF)

fill: bool = False

whether or not the polygon should be filled when drawn

legend: str = None

une légende optionnelle

points: List[*FPoint*]

List of points defining the polygon's shape

polygon_id: int = 0

identifiant unique

class sirilpy.models.**RegData**(*fwhm=0.0*, *weighted_fwhm=0.0*, *roundness=0.0*, *quality=0.0*, *background_lvl=0.0*, *number_of_stars=0*, *H=<factory>*)

Bases : object

Python equivalent of Siril regdata structure

classmethod deserialize(*data*)

Deserialize a binary response into a RegData object.

Paramètres
data (*bytes*) -- Données binaires à décompresser

Type renvoyé
RegData

Retourne : objet RegData

Raises : SirilError if the received data doesn't match the expected size'
struct.error Si le désarchivage échoue

H: *Homography*

Stores a homography matrix describing the affine transform from this frame to the reference frame

background_lvl: float32 = 0.0

niveau du fond de ciel

fwhm: float = 0.0

copy of fwhm->fwhmx, used as quality indicator

number_of_stars: int = 0

number of stars detected in the image

quality: float = 0.0

mesure de la qualité de l'image

roundness: float32 = 0.0

fwhm->fwhmy / fwhm->fwhmx, 0 when uninit,]0, 1] when set

weighted_fwhm: float32 = 0.0

utilisé pour exclure les images douteuses

class sirilpy.models.**Sequence**(*seqname=''*, *number=0*, *selnum=0*, *fixed=0*, *nb_layers=-1*, *rx=0*, *ry=0*, *is_variable=False*, *bitpix=0*, *reference_image=0*, *imgparam=None*, *regparam=None*, *stats=None*, *distoparam=None*, *beg=0*, *end=0*, *exposure=0.0*, *fz=False*, *type=None*, *cfa_opened_monochrome=False*, *current=0*)

Bases : object

Python equivalent of Siril sequ structure

beg: int = 0

imgparam[0]->filenum

bitpix: int = 0

format des pixels de l'image, à partir du fits

cfa_opened_monochrome: bool = False

CFA SER opened in monochrome mode

current: int = 0

numéro du fichier actuellement chargé

distoparam: List[*DistoData*] = None

distortion data for the sequence [nb_layers]

end: int = 0

imgparam[number-1]->filenum

exposure: float = 0.0

exposition de l'image

fixed: int = 0

longueur fixe de l'index d'image dans le nom de fichier

fz: bool = False
si le fichier est compressé
imgparam: List[*ImgData*] = None
a structure for each image of the sequence [number]
is_variable: bool = False
La séquence contient des images de différentes tailles
nb_layers: int = -1
nombre de couches intégrées dans chaque fichier image
number: int = 0
nombres d'images dans la séquence
reference_image: int = 0
image de référence pour l'alignement
regparam: List[List[*RegData*]] = None
registration parameters for each layer [nb_layers][number]
rx: int = 0
largeur de la première image
ry: int = 0
hauteur de la première image
selnum: int = 0
nombres d'images sélectionné
seqname: str = ''
nom de la séquence
stats: List[List[*ImageStats*]] = None
statistiques des images pour chaque couche [nb_couche][nombre]
type: *SequenceType* = None
le type de séquence

25.5 Sirilpy Enums

This module provides all the enums that are used within sirilpy. Enums submodule for Siril. This submodule contains all the enums used within sirilpy.

class sirilpy.enums.**BitpixType**(*value*)
Bases : IntEnum
Mimics the Siril bitpix enum. Note that although Siril can handle opening FITS files of any data type, internally it processes images only as USHORT_IMG (uint16) or FLOAT_IMG (float32).
BYTE_IMG = 8
DOUBLE_IMG = -64
FLOAT_IMG = -32
LONG_IMG = 32
SHORT_IMG = 16
USHORT_IMG = 20

class sirilpy.enums.**CommandStatus**(*value*)
Bases : IntEnum
Contains Siril command status codes, matching the values returned internally within Siril. These can be used for error handling. CMD_OK and CMD_NO_WAIT are no-error codes ; all the other codes represent command errors. These are available through the CommandError exception and may generally be handled without being regarded as fatal to the script.

CMD_ALLOC_ERROR = 1048576

CMD_ARG_ERROR = 32

CMD_DIR_NOT_FOUND = 4194304

CMD_FILE_NOT_FOUND = 131072

CMD_FOR_CFA_IMAGE = 65536

CMD_FOR_PLATE_SOLVED = 262144

CMD_GENERIC_ERROR = 128

CMD_IMAGE_NOT_FOUND = 256

CMD_INVALID_IMAGE = 1024

CMD_LOAD_IMAGE_FIRST = 2048

CMD_NEED_INIT_FIRST = 524288

CMD_NOT_FOR_MONO = 16384

CMD_NOT_FOR_RGB = 32768

CMD_NOT_FOR_SINGLE = 8192

CMD_NOT_FOUND = 1

CMD_NOT_SCRIPTABLE = 8

CMD_NO_CWD = 4

CMD_NO_WAIT = 2

CMD_OK = 0

CMD_ONLY_SINGLE_IMAGE = 4096

CMD_SELECTION_ERROR = 64

CMD_SEQUENCE_NOT_FOUND = 512

CMD_THREAD_RUNNING = 2097152

CMD_WRONG_N_ARG = 16

class sirilpy.enums.**DialogID**(*value*)

Bases : IntEnum

classmethod from_name(*name*)

Safely create from a str containing a potential dialog name (e.g. "ASINH_DIALOG", returning None if the name no longer exists.

Type renvoyé
DialogID | None

ABERRATION_DIALOG = 19

ABOUT_DIALOG = 0

ANNOTATE_DIALOG = 1

ASINH_DIALOG = 2

ASTROMETRY_DIALOG = 3

BACKGROUND_EXTRACTION_DIALOG = 4

BINXY_DIALOG = 6

CANON_FIXBANDING_DIALOG = 7

CCM_DIALOG = 8

```
CLAHE_DIALOG = 9
COLOR_CALIBRATION = 10
COMPOSITION_DIALOG = 12
COMPSTARS_DIALOG = 11
COSMETIC_DIALOG = 13
CURVES_DIALOG = 14
CWD_DIALOG = 15
DECONV_DIALOG = 5
DENOISE_DIALOG = 16
DIALOG_FFT = 17
DIALOG_STAR_REMIX = 18
EPF_DIALOG = 20
EXTRACT_CHANNEL_DIALOG = 21
EXTRACT_WAVELETS_LAYERS_DIALOG = 22
FILE_INFORMATION = 23
GHT_DIALOG = 24
HISTOGRAM_DIALOG = 25
ICC_DIALOG = 26
KEYWORDS_DIALOG = 27
LINEARMATCH_DIALOG = 28
MEDIAN_DIALOG = 29
MERGE_CFA_DIALOG = 30
NINA_LIGHT_CURVE = 31
OPEN_DIALOG = 32
PCC_DIALOG = 33
PIXEL_MATH_DIALOG = 34
PREFS_DIALOG = 35
RESAMPLE_DIALOG = 36
RGRADIENT_DIALOG = 37
ROTATION_DIALOG = 38
SATU_DIALOG = 40
SAVEAS_DIALOG = 41
SCNR_DIALOG = 42
SEQLIST_DIALOG = 43
SPLIT_CFA_DIALOG = 44
STARNET_DIALOG = 45
STARS_LIST_WINDOW = 46
STAT_WINDOW = 47
S_PCC_DIALOG = 39
```

UNPURPLE_DIALOG = 48

WAVELETS_DIALOG = 49

class sirilpy.enums.**DialogReq**(*value*)

Bases : IntEnum

Represents the requirements to open a dialog represented by a DialogID

ANY = 1

Image or sequence loaded

IMG = 2

Image loaded

MONO = 3

Mono image loaded

NONE = 4

No requirement, this action works at any time

PLTSOLVD = 5

Image or sequence frame loaded and plate solved

RGB = 6

RGB image loaded

RGBPLTSOLVD = 7

RGB image or sequence frame loaded and plate solved

SEQ = 8

Sequence loaded

class sirilpy.enums.**DistoType**(*value*)

Bases : IntEnum

Python equivalent of the Siril disto_source enum

DISTO_FILE = 2

Distorsion provenant d'un fichier fourni

DISTO_FILES = 4

Distortion stored in each file (true only from seq platesolve, even with no distortion, it will be checked upon reloading)

DISTO_FILE_COMET = 5

special for cometary alignement, to be detected by apply reg

DISTO_IMAGE = 1

Distorsion par rapport à l'image actuelle

DISTO_MASTER = 3

Distorsion à partir d'un fichier maître

DISTO_UNDEF = 0

Aucune distorsion

class sirilpy.enums.**ImageType**(*value*)

Bases : IntEnum

enum representing image functional types

BIAS = 4

DARK = 2

FLAT = 3

LIGHT = 1

UNKNOWN = 0

class sirilpy.enums.**LogColor**(*value*)

Bases : IntEnum

Defines colors available for use with SirilInterface.log() For consistency LogColor.Default should be used for normal messages, LogColor.Red should be used for error messages, LogColor.Salmon should be used for warning messages, LogColor.Green should be used for completion notifications, and LogColor.Blue should be used for technical messages such as equations, coefficients etc.

BLUE = 4

DEFAULT = 0

GREEN = 3

RED = 1

SALMON = 2

class sirilpy.enums.**PlotType**(*value*)

Bases : IntEnum

Enumeration of available plot types for visualizing data series.

HYPHENS = 2

LINES = 3

LINESHYPHENS = 6

LINESMARKS = 5

LINESPOINTS = 4

MARKS = 1

POINTS = 0

class sirilpy.enums.**STFType**(*value*)

Bases : IntEnum

enum représentant les types STF

ASINH_DISPLAY = 4

AUTOSTRETCH_DISPLAY = 5

HISTEQ_DISPLAY = 6

LINEAR_DISPLAY = 0

LOG_DISPLAY = 1

SQRT_DISPLAY = 2

SQUARED_DISPLAY = 3

class sirilpy.enums.**SequenceType**(*value*)

Bases : IntEnum

Python equivalent of the Siril sequence_type enum

SEQ_AVI = 3

SEQ_FITSEQ = 2

SEQ_INTERNAL = 4

SEQ_REGULAR = 0

SEQ_SER = 1

class sirilpy.enums.**SirilVport**

Bases : object

Définit les fenêtres d'affichage Siril

BLUE = 2

GREEN = 1

MONO = 0

RED = 0

RGB = 3

class sirilpy.enums.**SlidersMode**(*value*)

Bases : IntEnum

MINMAX = 1

MIPSLOHI = 0

USER = 2

class sirilpy.enums.**StarProfile**(*value*)

Bases : IntEnum

Python equivalent of the Siril starprofile enum. Used to identify the type of fit used to model a star in the image. Note that MOFFAT_FIXED is currently not used in Siril, but is reserved for future use for Moffat stars modelled with a fixed beta parameter

GAUSSIAN = 0

MOFFAT = 1

MOFFAT_FIXED = 2

25.6 Sirilpy Plotting

This submodule provides classes and method to access the native Siril plotting functionality. Of course, you can also use matplotlib but this submodule provides access to the same plotting capabilities as used internally within Siril for a more seamless result. All its members are made available at the module root level, there is no need to import models separately.

Once populated, the PlotData object can be plotted using `SirilInterface.xy_plot()`. Plot submodule for Siril, providing classes for plot data representation and serialization. This submodule enables users to create and configure various types of plots with customizable appearance and error bars.

class sirilpy.plot.**PlotData**(*title='Data Plot'*, *xlabel='X'*, *ylabel='Y'*, *savename='plot'*, *show_legend=True*, *datamin=None*, *datamax=None*)

Bases : object

Metadata container for plot configuration. The actual series data are held in SeriesData objects and can be added using the Class methods add_series or add_series_obj after initialization of the PlotData.

Membres :

title : Plot title xlabel : X-axis label ylabel : Y-axis label savename : Save filename (extension is added automatically) show_legend : bool indicating whether to show legend datamin : List [xmin, ymin] forcing the bottom left coordinate to show. If omitted, the range is set to the data range. datamax : List [xmax, ymax] forcing the top right coordinate to show. If omitted, the range is set to the data range.

classmethod serialize(*plot_data*)

Serialize plot data for shared memory transfer using network byte order.

Paramètres

plot_data (*PlotData*) -- Objet PlotData contenant la configuration du graphique

Type renvoyé
Tuple[bytes, int]

Renvoie
Tuple of serialized bytes and total length

add_series(*x_coords*, *y_coords*, *label=None*, *plot_type=PlotType.LINES*, *n_error=None*, *p_error=None*)

Add a new series to the plot metadata.

Renvoie
the created SeriesData object for further manipulation if needed.

Type renvoyé
SeriesData

add_series_obj(*series*)

Add a pre-created SeriesData object to the plot metadata.

Retourne : None

Type renvoyé
None

class sirilpy.plot.**SeriesData**(*x_coords*, *y_coords*, *label=None*, *plot_type=PlotType.LINES*, *n_error=None*, *p_error=None*)

Bases : object

Représente une seule série de données à tracer.

Membres :

x_coords : Either a List[float] or a np.ndarray containing the values for the x coordinates for this series

y_coords : Either a List[float] or a np.ndarray containing the values for the y coordinates for this series

label : A str containing a label for the series (shown in the plot legend)

plot_type : a PlotType setting the type of marks to use

n_error : Either a List[float] or a np.ndarray containing values for the y-axis negative errors for this series

p_error : Either a List[float] or a np.ndarray containing values for the y-axis positive errors for this series

25.7 Méthodes utilitaires Sirilpy

This submodule provides utility methods for use in Siril python scripts. The most important is `ensure_installed()` but there are also methods such as `download_with_progress()` which provides a highly robust method for downloading large files with good error recovery through a retries and resume mechanism. Utility module for Siril Python interface providing helper functions for file operations, package management, and standard I/O control to support Siril's scripting capabilities.

class sirilpy.utility.**SuppressedStderr**

Bases : object

This context manager allows suppression of the script's stderr, which can be useful if you are using module functions that are known to produce warnings that you want to avoid distracting the user with, such as FutureWarnings of features that have become deprecated but are in a dependency rather than your own code. The class should be used sparingly and should **not** be used to hide evidence of bad code.

class sirilpy.utility.**SuppressedStdout**

Bases : object

This context manager allows suppression of the script's stdout, which can be useful to avoid flooding the log with stdout messages from an excessively verbose module used in the script.

Exemple

```
import sirilpy as s
siril = s.SirilInterface()
print("This message will appear in the Siril log")
with s.SuppressedStdout():
    print("This message will not appear")
print("This message will appear again")
```

sirilpy.utility.**check_module_version**(*requires=None*)

Check the version of the Siril module is sufficient to support the script. This is not mandatory if you are only using classes, methods etc. that are provided in the initial public release, but if you rely on methods that are noted int he API documentation as having been added at a particular version of the module then you must check the running sirilpy module supports your script by calling this function.

Siril will not show scripts with unsatisfied check_module_version() calls in the repository list in 'Get Scripts', so if you want a function that enables different code paths for different versions you should use *needs_module_version()* instead.

Paramètres

requires (*str*) -- A version format specifier string following the same format used by pip, i.e. it may contain '==1.2', ' !=3.4', '>5.6', '>=7.8', or a combination such as '>=1.2,<3.4'

Renvoie

True if requires = None or if the available sirilpy module version satisfies the version specifier, otherwise False

Lève

ValueError -- if requires is an invalid version specifier.

sirilpy.utility.**download_with_progress**(*siril*, *url*, *file_path*, *max_retries=3*, *retry_delay=5*, *resume=True*)

Robust file download method with native Siril progress tracking and error handling using retries and a resume mechanism.

Type renvoyé

bool

sirilpy.utility.**ensure_installed**(**packages*, *version_constraints=None*, *reinstall=False*)

Ensures that the specified package(s) are installed and meet optional version constraints.

Paramètres

- ***packages** (*str or List[str]*) -- Name(s) of the package(s) to ensure are installed.
- **version_constraints** (*str or List[str], optional*) -- Version constraint string(s) (e.g. ">=1.5", "==2.0"). Can be a single constraint or a list matching packages.
- **reinstall** (*bool, optional*) -- Force la réinstallation. La valeur par défaut est False.

Renvoie

Vrai si tous les paquets sont installés avec succès ou répondent déjà aux contraintes.

Type renvoyé

bool

Lève

- *SirilError* -- Si l'installation du paquet échoue.
- **ValueError** -- Si le nombre de contraintes diffère du nombre de paquets à installer.
- **TimeoutError** -- Si pip échoue avec un délai d'attente apparent.
- **SystemExit** -- If a package that is already imported needs to be reinstalled with a different version.

sirilpy.utility.**human_readable_size**(*bytes_size*)

Convertir les bytes en un format lisible par l'homme.

Paramètres
bytes_size (*int*) -- Taille en bytes

Renvoie
Taille formatée avec l'unité appropriée (B, KB, MB, GB, TB)

Type renvoyé
str

Lève
TypeError -- sur un type d'entrée incorrect

sirilpy.utility.**needs_module_version**(*requires=None*)

Check the version of the Siril module is sufficient to support a feature. This allows writing optional code paths for different sirilpy API levels. The function works the same as check_module_version() but the presence of an unsatisfied needs_module_version() call will not prevent a script from showing up in the list in 'Get Scripts'

Paramètres
requires (*str*) -- A version format specifier string following the same format used by pip, i.e. it may contain '==1.2', ' !=3.4', '>5.6', '>=7.8', or a combination such as '>=1.2,<3.4'

Renvoie
True if requires = None or if the available sirilpy module version satisfies the version specifier, otherwise False

Lève
ValueError -- if requires is an invalid version specifier.

sirilpy.utility.**parse_fits_header**(*header_text*, *include_comments=False*)

Parse FITS header from text content into a dictionary with support for HIERARCH and CONTINUE keywords.

Handles Siril's newline-separated FITS header format and converts it to a dictionary compatible with astropy.wcs.WCS. Supports extended FITS keywords via HIERARCH and long string values via CONTINUE.

Paramètres
— **header_text** (str) -- Raw header string with header cards separated by newlines. Expected format : "KEYWORD = value / comment"
— **include_comments** (bool) -- If True, includes COMMENT and HISTORY cards in the output. If False (default), these cards are skipped.

Type renvoyé
Dict[str, Union[str, int, float, bool]]

Renvoie
Dictionary mapping FITS header keywords (str) to their parsed values. Values are converted to appropriate Python types : - 'T'/'F' -> bool - Quoted strings -> str (quotes removed, with CONTINUE support for long strings) - Numeric strings -> int or float - Everything else -> str - COMMENT/HISTORY cards -> str (content after keyword, if include_comments=True) - HIERARCH keywords -> str (full hierarchical keyword preserved)

Notes

— Filters out warning messages, tracebacks, and other non-header content
— By default ignores COMMENT and HISTORY cards (set include_comments=True to include)
— Skips malformed cards or invalid keywords
— Standard keywords must be 8 characters and alphanumeric (plus underscore/hyphen)
— HIERARCH keywords can be longer and contain spaces/dots
— CONTINUE cards are automatically merged with the previous string value
— Comments after '/' are ignored for regular cards

Exemple

```
>>> header_str = '''SIMPLE  = T / file conforms to FITS standard
... BITPIX  = -32 / bits per pixel
... HIERARCH ESO DET CHIP1 NAME = 'CCD #1' / detector name
... LONGSTR = 'This is a very long string that needs to be '
... CONTINUE  'continued on the next line'
... COMMENT Test comment'''
>>> result = parse_fits_header(header_str)
>>> result['SIMPLE']
True
>>> result['BITPIX']
-32
>>> result['HIERARCH ESO DET CHIP1 NAME']
'CCD #1'
>>> result['LONGSTR']
'This is a very long string that needs to be continued on the next line'
```

sirilpy.utility.**safe_subprocess_Popen**(*command*, ***kwargs*)

Provides a safe alternative to subprocess.Popen, accounting for the fact that a subprocess command line may be run from a flatpak environment and require use of flatpak-spawn to ensure it runs outside the sandbox, avoiding issues with subprocesses that rely on environment variables and so on. safe_subprocess_Popen is a drop-in replacement for subprocess.Popen for use in Siril scripts.

Introduit dans la version 1.0.3 de Sirilpy

sirilpy.utility.**safe_subprocess_run**(*command*, ***kwargs*)

Provides a safe alternative to subprocess.Popen, accounting for the fact that a subprocess command line may be run from a flatpak environment and require use of flatpak-spawn to ensure it runs outside the sandbox, avoiding issues with subprocesses that rely on environment variables and so on. safe_subprocess_run is a drop-in replacement for subprocess.run for use in Siril scripts.

Introduit dans la version 1.0.3 de Sirilpy

sirilpy.utility.**truncate_utf8**(*data*, *max_bytes*)

Tronque l'entrée utf8. Accepte soit des octets, soit des chaînes de caractères en entrée et renvoie les données dans le même format que l'entrée.

Paramètres
data (*bytes or str*) -- Les données à tronquer

Renvoie
Les données tronquées

Type renvoyé
bytes or str

Lève
TypeError -- si l'entrée n'est pas de type bytes ou str

sirilpy.utility.**uninstall_package**(*package_name*)

Uninstall a package using pip, streaming output to stdout. Introduced in sirilpy 1.0.15.

Paramètres
package_name (*str*) -- Name of the package to uninstall.

Lève
subprocess.CalledProcessError -- Si la désinstallation avec pip échoue.

25.8 Assistants GPU Sirilpy

This submodule provides helper classes to make it easier to manage GPU framework packages such as ONNX, Torch etc. for use in Siril python scripts. The landscape of these frameworks' support for different GPU architectures on different OSes is rapidly developing and the helpers aim to suggest reliable packages / configurations. This means that the proposed configurations may in some cases be conservative : the aim is to provide a good and robust level of GPU support for as many users as possible in an automated python environment rather than the absolute best, but perhaps fragile, optimisation. GPU helper module for Siril Python interface providing helper functions for detection, installation and testing of GPU-related modules. Initial scope is ONNX, torch and jax

class sirilpy.gpuhelper.**JaxHelper**

Bases : `object`

Une classe d'aide pour détecter, installer et tester JAX avec l'accélération matérielle appropriée.

Cette classe détecte automatiquement la configuration du système et installe la variante JAX appropriée (CPU, CUDA, ROCm, etc.) en fonction du matériel disponible.

ensure_jax()

Wrapper for install_jax() that only installs it if needed, with negligible overhead if it is already installed.

Type renvoyé
`bool`

get_recommended_backend()

Determine the recommended JAX backend and installation parameters. When multiple GPUs are present, uses priority system to choose the best one.

JAX has limited platform support : - NVIDIA : CUDA on Linux (Windows support experimental/limited) - AMD ROCm : Linux only - Apple Silicon : Metal backend - Intel : Experimental plugin with dependency issues

Type renvoyé
`Dict[str, Any]`

Renvoie
Dict with 'backend', 'packages', 'extra_index_url' keys

install_jax(*force_variant=None*, *version_constraint=None*, *force_reinstall=False*)

Install JAX with the appropriate variant for the detected hardware. Use this instead of ensure_installed() to ensure that jax is installed correctly for the given hardware / OS

Paramètres
— **force_variant** (`Optional[str]`) -- Override auto-detection with specific variant (e.g., 'jax[cpu]')
— **version_constraint** (`Optional[str]`) -- Chaîne de contrainte de version (par exemple, '>=0.4.0')

force_reinstall : Force une réinstallation

Renvoie
True if installation succeeded, False otherwise

Type renvoyé
bool

is_jax_installed()

Vérifie si PyTorch est installé sans l'importer.

Type renvoyé
`bool`

status()

Prints the current status of the Jax Helper class in regard to its support for different OSes, GPUs. The world of heterogenous computing is developing rapidly and while support for some of the frameworks for which helpers are available is not yet universally available, hopefully it will improve in the future.

test_jax()

Test JAX functionality and return execution provider.

Renvoie
the bool returned is True if jax works or False if it does not, and the str is "gpu" if JAX is using GPU, "cpu" if using CPU or None if

Type renvoyé
Tuple[bool,str]

Lève
— **RuntimeError** -- Si le calcul JAX échoue ou si la vérification de précision échoue
— **ImportError** -- Si JAX n'est pas installé

uninstall_jax(*dry_run=False*)

Detect and uninstall any existing JAX-related packages.

This is useful when you need to clean up a problematic JAX installation before installing a different variant (e.g., falling back from GPU to CPU).

Paramètres
dry_run (bool) -- Si True, détecter uniquement les paquets sans les désinstaller

Type renvoyé
Dict[str, Any]

Renvoie
Dictionnaire contenant des informations sur les paquets détectés et désinstallés

class sirilpy.gpuhelper.**ONNXHelper**

Bases : object

A class to handle detection and installation of the appropriate ONNX Runtime package based on the system hardware and configuration.

Example usage (this should be used instead of `sirilpy.ensure_installed("onnxruntime")` to install the correct package for the user's system.)

```
oh = sirilpy.ONNXHelper()
oh.ensure_onnxruntime()
```

ensure_onnxruntime()

Wrapper for install_onnxruntime() that only installs it if needed, with negligible overhead if it is already installed.

Type renvoyé
bool

get_execution_providers_ordered(*ai_gpu_acceleration=True*, *force_check=False*)

Get execution providers ordered by priority. This function returns a list of available ONNX Runtime execution providers in a reasonable order of priority, covering major GPU platforms : The CPU provider is always included as the final fallback option.

Paramètres
— **ai_gpu_acceleration** (*bool*) -- Indique s'il faut inclure les fournisseurs d'accélération GPU. La valeur par défaut est True.
— **force_check** (*bool*) -- Détermine s'il faut forcer une nouvelle vérification même si une configuration est mise en cache. La valeur par défaut est False.

Renvoie
Liste triée des fournisseurs d'exécution disponibles.

Type renvoyé
list

get_recommended_backend()

Determine the recommended ONNX Runtime backend based on hardware. When multiple GPUs are present, uses priority system to choose the best one.

Note : ONNX Runtime has platform-specific backend support : - Windows : DirectML for all GPUs (NVIDIA, AMD, Intel) - more reliable, no driver dependencies - Linux : CUDA for NVIDIA, CPU for AMD (no ROCm support yet), OpenVINO for Intel

Renvoie
'gpu', 'directml', 'openvino', 'coreml', ou 'cpu'

Type renvoyé
Backend name

import_onnxruntime()

Import onnxruntime, add it to the global dict, test if it's built against CUDA and if so preload the CUDA and CUDNN libraries to improve the chances of finding them if Torch[CUDA] happens to be installed.

install_onnxruntime(*force=False*)

Détecter la configuration du système et installer le package ONNX Runtime approprié.

Paramètres
force -- bool : Si True, force la réinstallation même si onnxruntime est déjà installé.

Renvoie
True if installation was successful or already installed, False otherwise.

Type renvoyé
bool

Lève
TimooutError -- if a TimeoutError occurs in ensure_installed() - this avoids falling back to the CPU-only package purely because of network issues

is_onnxruntime_installed()

Vérifiez si un paquet onnxruntime est déjà installé et opérationnel.

Renvoie

(is_installed, package_name) where package_name could be
'onnxruntime', 'onnxruntime-gpu', 'onnxruntime-silicon', etc.

Type renvoyé
tuple

run(*session*, *model_path*, *output_names*, *input_feed*, *run_options=None*, *return_first_output=False*)

Lancer l'inférence avec basculement automatique vers le CPU en cas d'échec de la session.

Paramètres
— **session** -- The ONNX runtime inference session
— **model_path** (*str*) -- Path to the ONNX model file (needed for CPU fallback)
— **output_names** -- Noms des sorties à calculer, ou None pour toutes les sorties
— **input_feed** -- Dictionnaire associant les noms d'entrée aux tenseurs d'entrée
— **run_options** -- Options d'exécution facultatives pour la session d'inférence
— **return_first_output** (*bool*) -- If True, return only the first output instead of the full list

Renvoie

(result, session) where result is the inference output (or first output if return_first_output=True) and
session is the (potentially updated) inference session

Type renvoyé
tuple

`status`()

Prints the current status of the ONNX Helper class in regard to its support for different OSes, GPUs. The world of heterogenous computing is developing rapidly and while support for some of the frameworks for which helpers are available is not yet universally available, hopefully it will improve in the future.

`test_onnxruntime`(*ort=None*)

Tester un fichier onnxruntime importé.

Args :install_torch(cuda_version=cuda_version)

ort : The ONNX runtime module to test. If None, the method will
attempt to import onnxruntime for the test.

Renvoie
a list of confirmed working ONNXRuntime ExecutionProviders in priority order

Type renvoyé
list

`uninstall_onnxruntime`()

Detects and uninstalls all variants of onnxruntime packages. Checks for any package starting with 'onnxruntime'.

Renvoie
Liste des paquets désinstallés

Type renvoyé
list

`class sirilpy.gpuhelper.TorchHelper`

Bases : `object`

Helper class for PyTorch detection, installation and testing.

`ensure_torch`(*cuda_version=None*)

Ensure PyTorch is installed with the appropriate backend.

Paramètres
`cuda_version` (`Optional[str]`) -- Optional CUDA version to override auto-detection (e.g., 'cu118', 'cu126', 'cu128')

Type renvoyé
`bool`

Retourne : True en cas de succès, False dans le cas contraire

`get_recommended_backend`()

Determine the recommended PyTorch backend and installation parameters. When multiple GPUs are present, uses priority system to choose the best one.

PyTorch has good cross-platform support : - NVIDIA : CUDA on all platforms - AMD ROCm-compatible : ROCm on Linux AND Windows - Intel Arc : XPU backend - Apple Silicon : MPS backend

Type renvoyé
`Dict[str, Any]`

Renvoie
Dictionnaire avec les clé 'backend', 'cuda_version', 'extra_index_url', 'packages'

`get_torch_device`(*use_gpu=True*)

Obtains a suitable torch device based on the capabilities of the installed torch package. if use_gpu is False, torch.device("cpu") will be returned. This function is available since sirilpy 1.0.17

Type renvoyé
torch.device

install_torch(*cuda_version=None*, *extra_index_url=None*, *packages=None*)

Install PyTorch with specified configuration.

Paramètres
- **cuda_version** (Optional[str]) -- CUDA version (e.g., 'cu118', 'cu126', 'cu128')
- **extra_index_url** (Optional[str]) -- PyTorch wheel repository URL
- **packages** (Optional[list]) -- Liste des paquets à installer

is_torch_installed()

Vérifie si PyTorch est installé sans l'importer.

Type renvoyé
bool

status()

Prints the current status of the Torch Helper class in regard to its support for different OSes, GPUs. The world of heterogenous computing is developing rapidly and while support for some of the frameworks for which helpers are available is not yet universally available, hopefully it will improve in the future.

test_torch()

Run tests to verify that torch is installed and runs correctly.

uninstall_torch()

Detects and uninstalls PyTorch and related packages. Checks for torch ecosystem packages. :returns : A list of uninstalled packages :rtype : list

sirilpy.gpuhelper.**detect_gpu_capabilities**()

Comprehensive GPU detection for all supported hardware.

Type renvoyé
Dict[str, Any]

Renvoie
Dict containing detected hardware information and recommendations for PyTorch, ONNX, and JAX backends.

sirilpy.gpuhelper.**get_gpu_detection_report**()

Generate a comprehensive GPU detection report suitable for display in a GUI.

Type renvoyé
Dict[str, Any]

Renvoie
Dict containing all detected hardware and recommended configurations for ONNX, PyTorch, and JAX, including GPU priority information.

25.9 Sirilpy TK GUI

Avertissement

This submodule is now deprecated. There are no longer any scripts in the repository that use tksiril and submissions to the siril-scripts repository will no longer be accepted if they use tksiril. You should migrate any personal scripts to use PyQt6 instead. **The tksiril submodule will be removed in a future version of Siril.**

This submodule provides some helper functions to support consistent GUI implementation using `tkinter`. It must be explicitly imported in order to be used. Note that when writing TKinter GUIs you should import ThemedTK from the ttkthemes module, because the bare TKinter UI shows up poorly on MacOS. ThemedTK and the methods available in the `tksiril` module help to provide a consistent look for Siril script GUIs on all platforms.

Avertissement

Linux users running a Wayland desktop should note that Tkinter does not yet have support for pure Wayland. In order to use python scripts utilising Tkinter GUIs you **must** have the XWayland compatibility package installed. If you don't, you will see errors about DISPLAY being unavailable.

```
from sirilpy import tksiril
```

TKsiril submodule for Siril, providing utility methods to achieve consistent script GUI appearance using the TKinter toolkit.

DEPRECATION WARNING : the tksiril submodule is deprecated from Siril 1.4.3. It will remain part of sirilpy throughout the stable 1.4 series but has been removed from master and will therefore disappear with the release of Siril 1.6.0.

No new scripts will be accepted in the siril-scripts repository if they use the tksiril submodule.

class sirilpy.tksiril.**ScrollableFrame**(*container*, **args*, ***kwargs*)

Bases : Frame

A scrollable frame widget that can contain other widgets.

This class creates a frame with vertical scrolling capability using a Canvas widget and Scrollbar. It supports both scrollbar and mouse wheel scrolling across all platforms (Windows, Mac, Linux).

Utilisation :

scrollable = ScrollableFrame(parent) scrollable.pack(fill="both", expand=True)

Add widgets to scrollable.scrollable_frame label = ttk.Label(scrollable.scrollable_frame, text="Hello") label.pack()

Optionally bind mouse wheel to child widgets scrollable.add_mousewheel_binding(label)

add_mousewheel_binding(*widget=None*)

Add mouse wheel scrolling support to a widget and its children.

This method recursively binds mouse wheel events to the specified widget and all its child widgets. It uses platform detection to apply the appropriate event bindings for each operating system.

Paramètres

widget -- The tkinter widget to bind mouse wheel events to. The binding will be applied recursively to all children. If no widget is specified it will default to the ScrollableFrame itself.

Exemple

Add a complex widget to the scrollable frame frame = ttk.Frame(scrollable.scrollable_frame) label = ttk.Label(frame, text="Hello") button = ttk.Button(frame, text="Click me")

Bind mouse wheel to the entire widget hierarchy scrollable.add_mousewheel_binding(frame)

sirilpy.tksiril.**create_tooltip**(*widget*, *text*, *wrap_length=250*)

Créer une info-bulle pour un widget Tkinter donné.

Paramètres

- **widget** (*tk.Widget*) -- Le widget auquel associer l'info-bulle
- **text** (*str*) -- Texte de l'info-bulle à afficher
- **max_width** (*int, optional*) -- Largeur maximale de l'info-bulle. La valeur par défaut est 300.
- **wrap_length** (*int, optional*) -- Longueur à partir de laquelle le texte passe à la ligne. La valeur par défaut est 250.

Lève

TypeError -- If text is not a string or the provided widget is not a valid Tkinter widget

sirilpy.tksiril.**elevate**(*root*)

Raises the Tk root window to the top of the window stack. Useful after calls to sirilpy methods that present child windows of the main Siril window such as info_messagebox().

NOTE : For this to work on KDE desktops, focus-stealing prevention must be disabled.

sirilpy.tksiril.**match_theme_to_siril**(*themed_tk*, *s*, *on_top=False*)

Match the Tkinter theme to the Siril configuration and set the script dialog to have topmost status, meaning that it will remain in front of other non-topmost windows.

Paramètres

- **s** (SirilInterface) -- sirilpy.SirilInterface class to provide the Siril theme (light or dark) to match
- **themed_tk** (*ThemedTk*) -- Instance de ThemedTk à laquelle appliquer le thème
- **on_top** -- si la fenêtre du script doit toujours rester au premier plan par rapport aux autres fenêtres

Lève

- **TypeError** -- Si les arguments de l'entrée ne sont pas du bon type
- **ValueError** -- Si la configuration du thème n'est ni 0 ou 1
- **AttributeError** -- Si les méthodes requises ne sont pas disponibles
- **RuntimeError** -- Si des erreurs surviennent lors de l'installation ou de la configuration du thème

sirilpy.tksiril.**standard_style**()

Provide a standardised ttk style to allow consistent visual appearance between different Siril python scripts.

Paramètres

none

Lève

SirilError -- Si la création ou la configuration du style échoue

25.10 Sirilpy Tkfilebrowser

Avertissement

This submodule is now deprecated. There are no longer any scripts in the repository that use tkfilebrowser and submissions to the siril-scripts repository will no longer be accepted if they use tkfilebrowser. You should migrate any personal scripts to use PyQt6 instead. **The tkfilebrowser submodule will be removed in a future version of Siril.**

This submodule is a fork of tkfilebrowser. The fork addresses a bug in the original code where duplicate device entries could cause errors in generating the filebrowser, and ensures the code can be mantained as the upstream package was last updated several years ago.

Documentation for tkfilebrowser can be found here. Note that some compatibility improvements have been made in the version included in sirilpy :

- `sirilpy.tkfilebrowser.askdirectory()` has been added as an alias for `tkfilebrowser.askopendirname()` to maintain compatibility with `tk.filedialog.askdirectory()`.
- In filefilter specifications `tkfilebrowser` requires multiple extensions to be provided separated by | whereas filedialog requires space-separated extensions : `sirilpy.tkfilebrowser` has been adapted to accept either format.

This submodule serves the sole purpose of being a drop-in replacement for the standard Tk filedialog on Linux, as the standard Tk filedialog is horrible on Linux. It can be used as a replacement like this :

```
if sys.platform.startswith("linux"):
    import sirilpy.tkfilebrowser as filedialog
else:
    from tkinter import filedialog
```

25.11 Exceptions Sirilpy

This submodule provides some customized exceptions to support Siril-specific error handling. All its members are made available at the module root level, there is no need to import it separately.

The sirilpy exceptions policy is as follows :

— At a low level within sirilpy methods a variety of exception types may be raised (those shown below as well as exceptions raised from other modules such as struct.error). Internal exception types are all descended from SirilError and may therefore be caught using *except SirilError*. Other exception types are re-raised as a SirilError to show the method where they were generated, but the underlying error can still be seen either in a traceback or using the Exception `__cause__` property.

— Some error types are recoverable errors such as NoImageError, NoSequenceError and CommandError. These exception types can be handled at script level (for example by showing a warning dialog reminding the user to load an image).

— Other error types are typically not recoverable, such as SharedMemoryError or SirilConnectionError.

Exceptions submodule for Siril, providing exception classes for use in exception raising within the sirilpy module.

exception sirilpy.exceptions.**CommandError**(*message='Command execution failed'*, *status_code=CommandStatus.CMD_GENERIC_ERROR*)

Bases : *SirilError*

Raised when a command sent to Siril fails to execute properly. (Note : 'command' in this case refers to internal commands sent from the python module to the Siril python handler, not Siril commands of the type that might be entered in the Siril command entry.) The full set of command status codes is shown in the CommandStatus enum. These exceptions are often recoverable and should therefore be handled before generically handling other SirilError types that are considered fatal.

status_code

(CommandStatus) Indicates the status code returned by the Siril command. This may be used in error handlers to allow scripts to handle some types of command error and continue (e.g. by prompting a user intervention).

exception sirilpy.exceptions.**DataError**(*message='Error handling data'*)

Bases : *SirilError*

Raised when there are problems with data handling. This includes cases like :

— Données d'image invalides

— Erreurs de conversion des données

— Échecs d'allocation de mémoire

— Débordements du buffer

exception sirilpy.exceptions.**ImageDialogOpenError**(*message='Siril image dialog is open'*)

Bases : *SirilError*

Exception raised when an image processing dialog is open.

exception sirilpy.exceptions.**MouseModeError**(*message='Siril mouse mode error'*)

Bases : *SirilError*

Exception raised when Siril is in the wrong mouse mode.

exception sirilpy.exceptions.**NoImageError**(*message='No Siril image loaded'*)

Bases : *SirilError*

Raised when a method requires an image to be loaded but no image is loaded. These exceptions are often recoverable and should therefore be handled before generically handling other SirilError types that are considered fatal.

exception sirilpy.exceptions.**NoSequenceError**(*message='No Siril sequence loaded'*)

Bases : *SirilError*

Raised when a method requires a sequence to be loaded but no sequence is loaded. These exceptions are often recoverable and should therefore be handled before generically handling other SirilError types that are considered fatal.

exception sirilpy.exceptions.**ProcessingThreadBusyError**(*message='Siril processing thread is busy'*)

Bases : *SirilError*

Exception raised when the processing thread is already in use.

exception sirilpy.exceptions.**SharedMemoryError**(*message='Siril shared memory error'*)

Bases : *SirilError*

Raised when there are problems connecting to or communicating with Siril using shared memory.

SharedMemoryError is not raised directly but will be wrapped in a SirilError. It should generally be regarded as fatal and the script should shut down gracefully if possible or just stop.

exception sirilpy.exceptions.**SirilConnectionError**(*message='Failed to connect to Siril'*)

Bases : *SirilError*

Soulevé en cas de problèmes de connexion ou de communication avec Siril. Cela inclut les cas suivants :
— Siril n'est pas en fonctionnement
— Échecs de connexion au socket
— Erreurs de protocole de communication
— Déconnexions inattendues

SirilConnectionError is not raised directly but will be wrapped in a SirilError. It should generally be regarded as fatal and the script should shut down gracefully if possible or just stop.

exception sirilpy.exceptions.**SirilError**(*message='An error occurred'*)

Bases : `Exception`

Classe d'exception de base pour toutes les erreurs liées à Siril.

Toutes les autres exceptions Siril héritent de cette classe, ce qui permet de détecter facilement toute erreur liée à Siril à l'aide d'une seule clause except.

CHAPITRE 26

Comment signaler des problèmes

Si, malgré l'étude de la documentation et des tutoriels, vous rencontrez un comportement étrange, cette page vous guidera sur la marche à suivre. Gardez à l'esprit que si quelque chose ne fonctionne pas, cela ne signifie pas nécessairement qu'il y a un bug. Le plus souvent, les problèmes sont dus à une utilisation incorrecte. Dans ce cas, il est essentiel de fournir des informations précises sur le problème, ce qui nous aidera à déterminer s'il s'agit d'un bug ou d'un malentendu. Reportez-vous à la section *Envoyez nous des informations utiles* pour obtenir des conseils. En effet, le simple fait de dire que cela ne fonctionne pas ne nous aidera pas à trouver une solution. Nous ne sommes pas des devins et nous avons besoin d'informations pour identifier le problème et trouver une solution. Il est donc important que vous nous disiez ce que vous faisiez au moment où le problème s'est produit, quel système d'exploitation vous utilisez, quelle version de Siril vous utilisez et surtout les logs !

26.1 Vérifier les listes de modifications et de bugs

Tout d'abord, si vous pensez avoir trouvé un bug dans Siril, il se peut qu'il ait déjà été signalé (parfois littéralement des dizaines de fois). Nous vous demandons donc de vérifier d'abord, par exemple en regardant les changelogs, ou les tickets qui ont déjà été **ouverts**, et même **fermés**.

26.2 Envoyez-nous des informations utiles

Comme indiqué dans l'introduction, nous avons besoin d'informations utiles pour résoudre le problème :

1. Quel système d'exploitation utilisez-vous ? Siril peut se comporter très différemment sous Windows, Linux ou macOS, c'est pourquoi nous avons besoin de cette information. Soyez aussi précis que possible.
2. Quelle version de Siril utilisez-vous ? Et comment l'avez-vous obtenue ? Paquet téléchargé sur le site web ? Par l'intermédiaire d'un tiers ? Compilé par vous-même ? Là encore, soyez le plus précis possible.
3. Il est parfois utile de partager des captures d'écran. Cependant, s'il vous plaît **NE PRENEZ PAS VOS CAPTURES D'ÉCRAN AVEC VOTRE SMARTPHONE** - c'est illisible. Votre système d'exploitation est capable de faire des captures d'écran très facilement (Google peut vous aider) et Siril offre également une telle fonctionnalité (le bouton avec l'appareil photo). Enfin, les formats souhaités sont des formats d'images : `jpg`, `bmp` ou `png`, mais absolument pas `pdf`.
4. Envoyez-nous des logs. Idéalement, nous préférons les logs en anglais ! Il suffit d'aller dans les préférences du Siril et de changer la langue en anglais dans l'onglet*Interface utilisateur*. Par ailleurs, il existe deux types de

logs : celui affiché dans la console Siril, qui décrit les étapes effectuées par le logiciel et peut nous aider à déboguer, et les logs internes qui sont visibles lorsque Siril est exécuté à partir de la ligne de commande :

- Les premiers sont très utiles dans la plupart des cas et peuvent être exportés très facilement à l'aide du bouton situé en bas à droite de ceux-ci. Cela crée un fichier qui peut facilement nous être envoyé.
- Cependant, lorsque le logiciel crash (c'est-à-dire qu'il se ferme soudainement sans avertissement), vous devez lancer Siril à partir de la ligne de commande, en essayant de reproduire le crash et de récupérer les logs. Ici, la méthode dépend du système d'exploitation.
  - Microsoft Windows : Ouvrez une fenêtre cmd (tapez `cmd` dans la barre de recherche de Windows) et tapez ce qui suit :

    ```
    "C:\Program Files\Siril\bin\siril.exe" 2>&1 >output.log
    ```

    Ceci sauvegardera le fichier output.log dans le dossier où le terminal a été démarré (dans la plupart des cas dans le dossier `%USERPROFILE%`).
  - macOS : Si vous avez installé Siril dans le dossier Applications, comme cela est généralement recommandé, commencez par ouvrir l'application Terminal à partir du dossier Utilities dans Applications, puis copiez et collez la ligne suivante :

    ```
    /Applications/Siril.app/Contents/MacOS/siril > ~/Desktop/output.log 2>&1
    ```

    Après le crash, les logs seront disponibles sur le bureau dans le fichier `output.log`.
  - GNU/Linux : Lancez simplement Siril dans un terminal. Habituellement, le binaire est situé dans la variable `$PATH`, dans ce cas tapez :

    ```
    siril > output.log 2>&1
    ```

    est tout ce dont vous avez besoin pour rediriger les logs dans le fichier `output.log`. Le fichier output.log sera sauvegardé dans le dossier où le terminal a été démarré

5. Envoyez-nous votre image. Si vous trouvez votre image étrange, n'hésitez pas à la partager avec nous, généralement au format FITS. C'est toujours plus intéressant qu'une capture d'écran. Pour ce faire, vous devez utiliser un service d'échange de fichiers volumineux. Il en existe beaucoup, et nous pouvons vous suggérer WeTransfer, par exemple. Dans ce cas, téléchargez vos données sur WeTransfer et obtenez un lien de téléchargement à partager.

26.3 Comment nous contacter ?

Il y a plusieurs façons de nous contacter et de signaler un bug. La plus simple est de nous trouver sur le forum (anglais), ou celui-ci (français). Mais il est également possible d'ouvrir un ticket sur notre dépôt gitlab. Dans ce cas, vérifiez d'abord que le même ticket n'a pas déjà été ouvert. Il se peut même qu'il ait été fermé parce qu'il a été résolu, dans ce cas, une courte description avec un numéro de ticket sera affichée dans le Changelog. Pour visualiser le ticket (même un ticket fermé) et confirmer que vous rencontrez ou non le même problème, rendez-vous à l'adresse https://gitlab.com/free-astro/siril/-/issues/XXXX avec XXXX le numéro de ticket.

> **Astuce**
>
> Les utilisateurs et les développeurs de Siril étant de nationalités différentes, la langue utilisée pour signaler un bug doit être l'anglais.

Bibliographie

[Fischler1981] Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

[Valdes1995] Valdes, F. G., Campusano, L. E., Velasquez, J. D., & Stetson, P. B. (1995). FOCAS automatic catalog matching algorithms. Publications of the Astronomical Society of the Pacific, 107(717), 1119.

[Losson2010] Olivier Losson, Ludovic Macaire, Yanqin Yang. Comparison of color demosaicing methods. Advances in Imaging and Electron Physics, 2010, 162, pp.173-265, section 2.2.2. https://hal.science/hal-00683233/document

[FruchterHook1997] A. S. Fruchter and R. N. Hook. (1997) *A novel image reconstruction method applied to deep Hubble Space Telescope images.* Proc. S.P.I.E. vol. 3164. https://arxiv.org/abs/astro-ph/9708242

[Avila2015] R. J. Avila, A. Koekemoer, J. Mack, A. Fruchter. (2015) *Optimizing pixfrac in Astrodrizzle : An Example from the Hubble Frontier Fields.* https://www.stsci.edu/files/live/sites/www/files/home/hst/instrumentation/wfc3/documentation/instrument-science-reports-isrs/_documents/2015/WFC3-2015-04.pdf

[Peter2009] Peter J. Huber and E. Ronchetti (2009), Robust Statistics, 2nd Ed., Wiley

[ConejeroPI] Juan Conejero, ImageIntegration, Tutoriel Pixinsight

[Rosner1983] Rosner, B. (1983). *Percentage points for a generalized ESD many-outlier procedure*. Technometrics, 25(2), 165-172.

[Brown2007] Brown, M., & Lowe, D. G. (2007). *Automatic panoramic image stitching using invariant features*. International journal of computer vision, 74, 59-73.

[GaiaDR3] Vallenari, A., et al. "Gaia Data Release 3-Summary of the content and survey properties." Astronomy & Astrophysics 674 (2023) : A1. 99(613), 191.

[SWIRE] https://www.iasf-milano.inaf.it/~polletta/templates/swire_templates.html

[Stellar] https://www.eso.org/sci/facilities/paranal/decommissioned/isaac/tools/lib.html

[Wiki_HEALPIX] https://en.wikipedia.org/wiki/HEALPix

[Young1994] https://opg.optica.org/ao/viewmedia.cfm?uri=ao-33-6-1108&seq=0

[Anger2018] Anger, J., Facciolo, G., & Delbracio, M. (2018). *Estimating an image's blur kernel using natural image statistics, and deblurring it : an analysis of the Goldstein-Fattal method.* Image Processing On Line, 8, 282-304. https://doi.org/10.5201/ipol.2018.211

[Anger2019] Anger, J., Facciolo, G., & Delbracio, M. (2019). Blind image deblurring using the l0 gradient prior. Image processing on line, 9, 124-142. https://doi.org/10.5201/ipol.2019.243

[Goldstein2012] Goldstein, A., & Fattal, R. (2012, October). *Blur-kernel estimation from spectral irregularities.* In European Conference on Computer Vision (pp. 622-635). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33715-4_45

[Lucy1974] Lucy, L. B. (1974). *An iterative technique for the rectification of observed distributions.* The astronomical journal, 79, 745. https://doi.org/10.1086/111605.

[Lebrun2013] Lebrun, M., Buades, A., & Morel, J. M. (2013) *Implementation of the "Non-Local Bayes" (NL-Bayes) Image Denoising Algorithm.* Image Processing On Line, 3 , pp. 1–42. https://doi.org/10.5201/ipol.2013.16

[Pierazzo2017] Pierazzo, N., & Facciolo, G. (2017). *Data adaptive dual domain denoising : a method to boost state of the art denoising algorithms.* Image Processing On Line, 7, 93-114. https://doi.org/10.5201/ipol.2017.203

[Mäkitalo2011] Mäkitalo, M., & Foi, A. (2012, March). *Poisson-gaussian denoising using the exact unbiased inverse of the generalized anscombe transformation.* In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1081-1084). IEEE. https://doi.org/10.1109/ICASSP.2012.6288074

[Mäkitalo2012] Makitalo, M., & Foi, A. (2011). *A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation.* IEEE transactions on image processing, 20(9), 2697-2698. https://doi.org/10.1109/TIP.2011.2121085

[Romano2015] Romano, Y., & Elad, M. (2015). *Boosting of image denoising algorithms.* SIAM Journal on Imaging Sciences, 8(2), 1187-1219. https://doi.org/10.1137/140990978

[Wright2003] Wright, Grady Barrett. Radial basis function interpolation : numerical and analytical developments. University of Colorado at Boulder, 2003.

[Stetson1987] Stetson, P. B. (1987). DAOPHOT : A computer program for crowded-field stellar photometry. Publications of the Astronomical Society of the Pacific, 99(613), 191.

Index des modules Python

S

sirilpy.connection, 553
sirilpy.enums, 589
sirilpy.exceptions, 606
sirilpy.gpuhelper, 599
sirilpy.models, 574
sirilpy.plot, 594
sirilpy.tksiril, 604
sirilpy.utility, 595

Commands Index

A

addmax, 477
asinh, 478
autoghs, 478
autostretch, 478

B

bg, 479
bgnoise, 479
binxy, 479
boxselect, 479

C

calibrate, 480
calibrate_single, 480
capabilities, 481
catsearch, 481
ccm, 482
cd, 482
cdg, 482
clahe, 483
clear, 483
clearstar, 483
close, 483
conesearch, 483
convert, 484
convertraw, 485
cosme, 485
cosme_cfa, 485
crop, 486

D

ddp, 486
denoise, 486
dir, 487
disto, 488
dumpheader, 488

E

entropy, 488
epf, 488
exit, 489
extract, 489
extract_Green, 489
extract_Ha, 490
extract_HaOIII, 490

F

fdiv, 490
ffill, 490
fftd, 491
ffti, 491
fill, 491
find_cosme, 491
find_cosme_cfa, 492
find_hot, 492
findcompstars, 492
findstar, 493
fix_xtrans, 494
fixbanding, 494
fmedian, 494
fmul, 494

G

gauss, 495
get, 495
getref, 495
ght, 495
grey_flat, 496

H

help, 496
histo, 497

I

iadd, 497
icc_assign, 497
icc_convert_to, 497
icc_remove, 498

idiv, 498
imul, 498
inspector, 498
invght, 499
invmodasinh, 499
invmtf, 499
isub, 500

J

jsonmetadata, 500

L

light_curve, 500
limit, 501
linear_match, 501
link, 502
linstretch, 502
livestack, 502
load, 503
load_seq, 503
log, 503
ls, 503

M

makepsf, 504
merge, 505
merge_cfa, 505
mirrorx, 505
mirrorx_single, 506
mirrory, 506
modasinh, 506
mtf, 507

N

neg, 507
new, 507
nozero, 508

O

offline, 508
offset, 508
online, 508

P

parse, 509
pcc, 509
platesolve, 509
pm, 511
profile, 511
psf, 511
pwd, 512
pyscript, 512

R

register, 512
reloadscripts, 514
requires, 514
resample, 514
rgbcomp, 515
rgradient, 515
rl, 515
rmgreen, 516
rotate, 516
rotatePi, 516

S

satu, 517
save, 517
savebmp, 517
savejpg, 518
savejxl, 518
savepng, 518
savepnm, 519
savetif, 519
savetif32, 519
savetif8, 519
sb, 520
select, 520
seqapplyreg, 520
seqccm, 522
seqclean, 522
seqcosme, 523
seqcosme_cfa, 523
seqcrop, 523
seqextract_Green, 524
seqextract_Ha, 524
seqextract_HaOIII, 524
seqfind_cosme, 524
seqfind_cosme_cfa, 525
seqfindstar, 525
seqfixbanding, 525
seqght, 526
seqheader, 526
seqinvght, 526
seqinvmodasinh, 526
seqlinstretch, 527
seqmerge_cfa, 527
seqmodasinh, 527
seqmtf, 528
seqprofile, 528
seqpsf, 528
seqplatesolve, 529
seqresample, 530
seqrl, 530
seqsb, 531
seqsetmag, 531
seqsplit_cfa, 531

seqstarnet, 532
seqstat, 532
seqsubsky, 533
seqtilt, 533
sequnsetmag, 533
sequpdate_key, 533
seqwiener, 534
set, 534
set16bits, 534
set32bits, 535
setcompress, 535
setcpu, 535
setext, 535
setfindstar, 536
setmag, 537
setmem, 537
setphot, 537
setref, 538
show, 538
spcc, 538
spcc_list, 539
split, 540
split_cfa, 540
stack, 540
stackall, 542
starnet, 542
start_ls, 543
stat, 543
stop_ls, 543
subsky, 544
synthstar, 544

T

threshlo, 545
threshhi, 545
thresh, 545
tilt, 545
trixel, 546

U

unclipstars, 546
unpurple, 546
unselect, 547
unsetmag, 547
unsharp, 547
update_key, 547

V

visu, 548

W

wavelet, 548
wiener, 548
wrecons, 549

Index

A

A (*attribut sirilpy.models.PSFStar*), 584
A_err (*attribut sirilpy.models.PSFStar*), 584
ABERRATION_DIALOG (*attribut sirilpy.enums.DialogID*), 590
ABOUT_DIALOG (*attribut sirilpy.enums.DialogID*), 590
add_mousewheel_binding() (*méthode sirilpy.tksiril.ScrollableFrame*), 604
add_series() (*méthode sirilpy.plot.PlotData*), 595
add_series_obj() (*méthode sirilpy.plot.PlotData*), 595
addmax, **477**
airmass (*attribut sirilpy.models.FKeywords*), 578
airmass (*attribut sirilpy.models.ImgData*), 583
allocate_data() (*méthode sirilpy.models.FFit*), 575
analyse_image_from_file() (*méthode sirilpy.connection.SirilInterface*), 553
ang_err (*attribut sirilpy.models.PSFStar*), 584
angle (*attribut sirilpy.models.PSFStar*), 584
ANNOTATE_DIALOG (*attribut sirilpy.enums.DialogID*), 590
ANY (*attribut sirilpy.enums.DialogReq*), 592
aperture (*attribut sirilpy.models.FKeywords*), 578
asinh, **478**
ASINH_DIALOG (*attribut sirilpy.enums.DialogID*), 590
ASINH_DISPLAY (*attribut sirilpy.enums.STFType*), 593
ASTROMETRY_DIALOG (*attribut sirilpy.enums.DialogID*), 590
autoghs, **478**
autostretch, **478**
AUTOSTRETCH_DISPLAY (*attribut sirilpy.enums.STFType*), 593
avgDev (*attribut sirilpy.models.ImageStats*), 582

B

B (*attribut sirilpy.models.PSFStar*), 584
B_err (*attribut sirilpy.models.PSFStar*), 584
BACKGROUND_EXTRACTION_DIALOG (*attribut sirilpy.enums.DialogID*), 590
background_lvl (*attribut sirilpy.models.RegData*), 588
bayer_pattern (*attribut sirilpy.models.FKeywords*), 578
bayer_xoffset (*attribut sirilpy.models.FKeywords*), 578
bayer_yoffset (*attribut sirilpy.models.FKeywords*), 578
beg (*attribut sirilpy.models.Sequence*), 588
beta (*attribut sirilpy.models.PSFStar*), 584
beta_err (*attribut sirilpy.models.PSFStar*), 584
bg, **479**
bgnoise, **479**
bgnoise (*attribut sirilpy.models.ImageAnalysis*), 581
bgnoise (*attribut sirilpy.models.ImageStats*), 582
BGSample (*classe dans sirilpy.models*), 574
BIAS (*attribut sirilpy.enums.ImageType*), 592
binning_x (*attribut sirilpy.models.FKeywords*), 578
binning_y (*attribut sirilpy.models.FKeywords*), 578
binxy, **479**
BINXY_DIALOG (*attribut sirilpy.enums.DialogID*), 590
bitpix (*attribut sirilpy.models.FFit*), 576
bitpix (*attribut sirilpy.models.Sequence*), 588
BitpixType (*classe dans sirilpy.enums*), 589
BLUE (*attribut sirilpy.enums.LogColor*), 593
BLUE (*attribut sirilpy.enums.SirilVport*), 594
Bmag (*attribut sirilpy.models.PSFStar*), 584
boxselect, **479**
bscale (*attribut sirilpy.models.FKeywords*), 578
BV (*attribut sirilpy.models.PSFStar*), 584
BYTE_IMG (*attribut sirilpy.enums.BitpixType*), 589
bzero (*attribut sirilpy.models.FKeywords*), 578

C

calibrate, **480**
calibrate_single, **480**
CANON_FIXBANDING_DIALOG (*attribut sirilpy.enums.DialogID*), 590
capabilities, **481**
catsearch, **481**
ccd_temp (*attribut sirilpy.models.FKeywords*), 578
ccm, **482**

CCM_DIALOG (*attribut sirilpy.enums.DialogID*), 590
cd, **482**
cdg, **482**
centalt (*attribut sirilpy.models.FKeywords*), 578
centaz (*attribut sirilpy.models.FKeywords*), 578
cfa_opened_monochrome (*attribut sirilpy.models.Sequence*), 588
channels (*attribut sirilpy.models.ImageAnalysis*), 581
channels (*propriété sirilpy.models.FFit*), 576
check_module_version() (*dans le module sirilpy.utility*), 596
checksum (*attribut sirilpy.models.FFit*), 576
clahe, **483**
CLAHE_DIALOG (*attribut sirilpy.enums.DialogID*), 590
clear, **483**
clear_image_bgsamples() (*méthode sirilpy.connection.SirilInterface*), 553
clear_undo_history() (*méthode sirilpy.connection.SirilInterface*), 553
clearstar, **483**
close, **483**
cmd() (*méthode sirilpy.connection.SirilInterface*), 553
CMD_ALLOC_ERROR (*attribut sirilpy.enums.CommandStatus*), 589
CMD_ARG_ERROR (*attribut sirilpy.enums.CommandStatus*), 590
CMD_DIR_NOT_FOUND (*attribut sirilpy.enums.CommandStatus*), 590
CMD_FILE_NOT_FOUND (*attribut sirilpy.enums.CommandStatus*), 590
CMD_FOR_CFA_IMAGE (*attribut sirilpy.enums.CommandStatus*), 590
CMD_FOR_PLATE_SOLVED (*attribut sirilpy.enums.CommandStatus*), 590
CMD_GENERIC_ERROR (*attribut sirilpy.enums.CommandStatus*), 590
CMD_IMAGE_NOT_FOUND (*attribut sirilpy.enums.CommandStatus*), 590
CMD_INVALID_IMAGE (*attribut sirilpy.enums.CommandStatus*), 590
CMD_LOAD_IMAGE_FIRST (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NEED_INIT_FIRST (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NO_CWD (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NO_WAIT (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NOT_FOR_MONO (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NOT_FOR_RGB (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NOT_FOR_SINGLE (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NOT_FOUND (*attribut sirilpy.enums.CommandStatus*), 590
CMD_NOT_SCRIPTABLE (*attribut sirilpy.enums.CommandStatus*), 590
CMD_OK (*attribut sirilpy.enums.CommandStatus*), 590
CMD_ONLY_SINGLE_IMAGE (*attribut sirilpy.enums.CommandStatus*), 590
CMD_SELECTION_ERROR (*attribut sirilpy.enums.CommandStatus*), 590
CMD_SEQUENCE_NOT_FOUND (*attribut sirilpy.enums.CommandStatus*), 590
CMD_THREAD_RUNNING (*attribut sirilpy.enums.CommandStatus*), 590
CMD_WRONG_N_ARG (*attribut sirilpy.enums.CommandStatus*), 590
color (*attribut sirilpy.models.Polygon*), 587
COLOR_CALIBRATION (*attribut sirilpy.enums.DialogID*), 591
color_managed (*attribut sirilpy.models.FFit*), 576
command_error_message() (*méthode sirilpy.connection.SirilInterface*), 553
CommandError, 606
CommandStatus (*classe dans sirilpy.enums*), 589
COMPOSITION_DIALOG (*attribut sirilpy.enums.DialogID*), 591
COMPSTARS_DIALOG (*attribut sirilpy.enums.DialogID*), 591
conesearch, **483**
confirm_messagebox() (*méthode sirilpy.connection.SirilInterface*), 554
connect() (*méthode sirilpy.connection.SirilInterface*), 554
contains_point() (*méthode sirilpy.models.Polygon*), 586
convert, **484**
convertraw, **485**
cosme, **485**
cosme_cfa, **485**
COSMETIC_DIALOG (*attribut sirilpy.enums.DialogID*), 591
create_new_seq() (*méthode sirilpy.connection.SirilInterface*), 554
create_tooltip() (*dans le module sirilpy.tksiril*), 604
crop, **486**
current (*attribut sirilpy.models.Sequence*), 588
CURVES_DIALOG (*attribut sirilpy.enums.DialogID*), 591
cvf (*attribut sirilpy.models.FKeywords*), 578
CWD_DIALOG (*attribut sirilpy.enums.DialogID*), 591

D

DARK (*attribut sirilpy.enums.ImageType*), 592
data (*propriété sirilpy.models.FFit*), 576
data_max (*attribut sirilpy.models.FKeywords*), 578
data_min (*attribut sirilpy.models.FKeywords*), 578

DataError, 606
date (*attribut sirilpy.models.FKeywords*), 578
date_obs (*attribut sirilpy.models.FKeywords*), 578
date_obs (*attribut sirilpy.models.ImgData*), 583
ddp, **486**
dec (*attribut sirilpy.models.FKeywords*), 578
dec (*attribut sirilpy.models.PSFStar*), 584
DECONV_DIALOG (*attribut sirilpy.enums.DialogID*), 591
DEFAULT (*attribut sirilpy.enums.LogColor*), 593
denoise, **486**
DENOISE_DIALOG (*attribut sirilpy.enums.DialogID*), 591
deserialize() (*méthode de la classe sirilpy.models.BGSample*), 575
deserialize() (*méthode de la classe sirilpy.models.FKeywords*), 577
deserialize() (*méthode de la classe sirilpy.models.ImageAnalysis*), 581
deserialize() (*méthode de la classe sirilpy.models.ImageStats*), 582
deserialize() (*méthode de la classe sirilpy.models.ImgData*), 583
deserialize() (*méthode de la classe sirilpy.models.PSFStar*), 583
deserialize() (*méthode de la classe sirilpy.models.RegData*), 587
deserialize_polygon() (*méthode de la classe sirilpy.models.Polygon*), 586
deserialize_polygon_list() (*méthode de la classe sirilpy.models.Polygon*), 586
detect_gpu_capabilities() (*dans le module sirilpy.gpuhelper*), 603
DIALOG_FFT (*attribut sirilpy.enums.DialogID*), 591
DIALOG_STAR_REMIX (*attribut sirilpy.enums.DialogID*), 591
DialogID (*classe dans sirilpy.enums*), 590
DialogReq (*classe dans sirilpy.enums*), 592
dir, **487**
disconnect() (*méthode sirilpy.connection.SirilInterface*), 554
disto, **488**
DISTO_FILE (*attribut sirilpy.enums.DistoType*), 592
DISTO_FILE_COMET (*attribut sirilpy.enums.DistoType*), 592
DISTO_FILES (*attribut sirilpy.enums.DistoType*), 592
DISTO_IMAGE (*attribut sirilpy.enums.DistoType*), 592
DISTO_MASTER (*attribut sirilpy.enums.DistoType*), 592
DISTO_UNDEF (*attribut sirilpy.enums.DistoType*), 592
DistoData (*classe dans sirilpy.models*), 575
distoparam (*attribut sirilpy.models.Sequence*), 588
DistoType (*classe dans sirilpy.enums*), 592
DOUBLE_IMG (*attribut sirilpy.enums.BitpixType*), 589
download_with_progress() (*dans le module sirilpy.utility*), 596
dumpheader, **488**

E

elevate() (*dans le module sirilpy.tksiril*), 604
end (*attribut sirilpy.models.Sequence*), 588
ensure_data_type() (*méthode sirilpy.models.FFit*), 576
ensure_installed() (*dans le module sirilpy.utility*), 596
ensure_jax() (*méthode sirilpy.gpuhelper.JaxHelper*), 599
ensure_onnxruntime() (*méthode sirilpy.gpuhelper.ONNXHelper*), 600
ensure_torch() (*méthode sirilpy.gpuhelper.TorchHelper*), 602
entropy, **488**
epf, **488**
EPF_DIALOG (*attribut sirilpy.enums.DialogID*), 591
error_messagebox() (*méthode sirilpy.connection.SirilInterface*), 554
estimate_noise() (*méthode sirilpy.models.FFit*), 576
exit, **489**
expend (*attribut sirilpy.models.FKeywords*), 579
exposure (*attribut sirilpy.models.FKeywords*), 579
exposure (*attribut sirilpy.models.Sequence*), 588
expstart (*attribut sirilpy.models.FKeywords*), 579
extract, **489**
EXTRACT_CHANNEL_DIALOG (*attribut sirilpy.enums.DialogID*), 591
extract_Green, **489**
extract_Ha, **490**
extract_HaOIII, **490**
EXTRACT_WAVELETS_LAYERS_DIALOG (*attribut sirilpy.enums.DialogID*), 591

F

fdiv, **490**
ffill, **490**
FFit (*classe dans sirilpy.models*), 575
fftd, **491**
ffti, **491**
fhi (*attribut sirilpy.models.FKeywords*), 579
FILE_INFORMATION (*attribut sirilpy.enums.DialogID*), 591
filename (*attribut sirilpy.models.DistoData*), 575
filename (*attribut sirilpy.models.FKeywords*), 579
filenum (*attribut sirilpy.models.ImgData*), 583
fill, **491**
fill (*attribut sirilpy.models.Polygon*), 587
filter (*attribut sirilpy.models.FKeywords*), 579
filter (*attribut sirilpy.models.ImageAnalysis*), 581
find_cosme, **491**
find_cosme_cfa, **492**
find_hot, **492**
findcompstars, **492**
findstar, **493**

fix_xtrans, **494**
fixbanding, **494**
fixed (*attribut sirilpy.models.Sequence*), 588
FKeywords (*classe dans sirilpy.models*), 577
FLAT (*attribut sirilpy.enums.ImageType*), 592
FLEN (*attribut sirilpy.models.ImageAnalysis*), 581
FLEN_VALUE (*attribut sirilpy.models.FKeywords*), 578
flength (*attribut sirilpy.models.FKeywords*), 579
flo (*attribut sirilpy.models.FKeywords*), 579
FLOAT_IMG (*attribut sirilpy.enums.BitpixType*), 589
fmedian, **494**
fmul, **494**
focal_length (*attribut sirilpy.models.FKeywords*), 579
focname (*attribut sirilpy.models.FKeywords*), 579
foctemp (*attribut sirilpy.models.FKeywords*), 579
focuspos (*attribut sirilpy.models.FKeywords*), 579
focussz (*attribut sirilpy.models.FKeywords*), 579
FPoint (*classe dans sirilpy.models*), 580
from_name() (*méthode de la classe sirilpy.enums.DialogID*), 590
from_rectangle() (*méthode de la classe sirilpy.models.Polygon*), 586
fwhm (*attribut sirilpy.models.ImageAnalysis*), 581
fwhm (*attribut sirilpy.models.RegData*), 588
fwhmx (*attribut sirilpy.models.PSFStar*), 584
fwhmx_arcsec (*attribut sirilpy.models.PSFStar*), 584
fwhmy (*attribut sirilpy.models.PSFStar*), 584
fwhmy_arcsec (*attribut sirilpy.models.PSFStar*), 584
fz (*attribut sirilpy.models.Sequence*), 588

G

gain (*attribut sirilpy.models.FKeywords*), 579
gauss, **495**
GAUSSIAN (*attribut sirilpy.enums.StarProfile*), 594
get, **495**
get_bounds() (*méthode sirilpy.models.Polygon*), 586
get_channel() (*méthode sirilpy.models.FFit*), 576
get_execution_providers_ordered() (*méthode sirilpy.gpuhelper.ONNXHelper*), 600
get_gpu_detection_report() (*dans le module sirilpy.gpuhelper*), 603
get_image() (*méthode sirilpy.connection.SirilInterface*), 555
get_image_bgsamples() (*méthode sirilpy.connection.SirilInterface*), 555
get_image_filename() (*méthode sirilpy.connection.SirilInterface*), 555
get_image_fits_header() (*méthode sirilpy.connection.SirilInterface*), 556
get_image_history() (*méthode sirilpy.connection.SirilInterface*), 556
get_image_iccprofile() (*méthode sirilpy.connection.SirilInterface*), 556
get_image_keywords() (*méthode sirilpy.connection.SirilInterface*), 556
get_image_pixeldata() (*méthode sirilpy.connection.SirilInterface*), 557
get_image_shape() (*méthode sirilpy.connection.SirilInterface*), 557
get_image_stars() (*méthode sirilpy.connection.SirilInterface*), 557
get_image_stats() (*méthode sirilpy.connection.SirilInterface*), 558
get_image_unknown_keys() (*méthode sirilpy.connection.SirilInterface*), 558
get_max_x() (*méthode sirilpy.models.Polygon*), 587
get_max_y() (*méthode sirilpy.models.Polygon*), 587
get_min_x() (*méthode sirilpy.models.Polygon*), 587
get_min_y() (*méthode sirilpy.models.Polygon*), 587
get_recommended_backend() (*méthode sirilpy.gpuhelper.JaxHelper*), 599
get_recommended_backend() (*méthode sirilpy.gpuhelper.ONNXHelper*), 600
get_recommended_backend() (*méthode sirilpy.gpuhelper.TorchHelper*), 602
get_selection_star() (*méthode sirilpy.connection.SirilInterface*), 558
get_selection_stats() (*méthode sirilpy.connection.SirilInterface*), 559
get_seq() (*méthode sirilpy.connection.SirilInterface*), 559
get_seq_distodata() (*méthode sirilpy.connection.SirilInterface*), 559
get_seq_frame() (*méthode sirilpy.connection.SirilInterface*), 559
get_seq_frame_filename() (*méthode sirilpy.connection.SirilInterface*), 560
get_seq_frame_header() (*méthode sirilpy.connection.SirilInterface*), 560
get_seq_frame_pixeldata() (*méthode sirilpy.connection.SirilInterface*), 560
get_seq_imgdata() (*méthode sirilpy.connection.SirilInterface*), 561
get_seq_regdata() (*méthode sirilpy.connection.SirilInterface*), 561
get_seq_stats() (*méthode sirilpy.connection.SirilInterface*), 561
get_siril_active_vport() (*méthode sirilpy.connection.SirilInterface*), 562
get_siril_config() (*méthode sirilpy.connection.SirilInterface*), 562
get_siril_configdir() (*méthode sirilpy.connection.SirilInterface*), 562
get_siril_display_iccprofile() (*méthode sirilpy.connection.SirilInterface*), 563
get_siril_log() (*méthode sirilpy.connection.SirilInterface*), 563

get_siril_panzoom() (*méthode sirilpy.connection.SirilInterface*), 563
get_siril_selection() (*méthode sirilpy.connection.SirilInterface*), 563
get_siril_slider_state() (*méthode sirilpy.connection.SirilInterface*), 563
get_siril_stf() (*méthode sirilpy.connection.SirilInterface*), 563
get_siril_stf_linked() (*méthode sirilpy.connection.SirilInterface*), 564
get_siril_systemdatadir() (*méthode sirilpy.connection.SirilInterface*), 564
get_siril_userdatadir() (*méthode sirilpy.connection.SirilInterface*), 564
get_siril_wd() (*méthode sirilpy.connection.SirilInterface*), 564
get_torch_device() (*méthode sirilpy.gpuhelper.TorchHelper*), 602
getref, **495**
ght, **495**
GHT_DIALOG (*attribut sirilpy.enums.DialogID*), 591
GREEN (*attribut sirilpy.enums.LogColor*), 593
GREEN (*attribut sirilpy.enums.SirilVport*), 594
grey_flat, **496**

H

H (*attribut sirilpy.models.RegData*), 588
h00 (*attribut sirilpy.models.Homography*), 581
h01 (*attribut sirilpy.models.Homography*), 581
h02 (*attribut sirilpy.models.Homography*), 581
h10 (*attribut sirilpy.models.Homography*), 581
h11 (*attribut sirilpy.models.Homography*), 581
h12 (*attribut sirilpy.models.Homography*), 581
h20 (*attribut sirilpy.models.Homography*), 581
h21 (*attribut sirilpy.models.Homography*), 581
h22 (*attribut sirilpy.models.Homography*), 581
has_saturated (*attribut sirilpy.models.PSFStar*), 584
header (*attribut sirilpy.models.FFit*), 576
height (*attribut sirilpy.models.ImageAnalysis*), 581
height (*propriété sirilpy.models.FFit*), 576
help, **496**
hi (*attribut sirilpy.models.FKeywords*), 579
HISTEQ_DISPLAY (*attribut sirilpy.enums.STFType*), 593
histo, **497**
HISTOGRAM_DIALOG (*attribut sirilpy.enums.DialogID*), 591
history (*attribut sirilpy.models.FFit*), 576
Homography (*classe dans sirilpy.models*), 580
human_readable_size() (*dans le module sirilpy.utility*), 596
HYPHENS (*attribut sirilpy.enums.PlotType*), 593

I

iadd, **497**
icc_assign, **497**
icc_convert_to, **497**
ICC_DIALOG (*attribut sirilpy.enums.DialogID*), 591
icc_profile (*propriété sirilpy.models.FFit*), 577
icc_remove, **498**
idiv, **498**
image_lock() (*méthode sirilpy.connection.SirilInterface*), 564
image_type (*attribut sirilpy.models.FKeywords*), 579
ImageAnalysis (*classe dans sirilpy.models*), 581
ImageDialogOpenError, 606
ImageStats (*classe dans sirilpy.models*), 582
imagetype (*attribut sirilpy.models.ImageAnalysis*), 581
ImageType (*classe dans sirilpy.enums*), 592
IMG (*attribut sirilpy.enums.DialogReq*), 592
ImgData (*classe dans sirilpy.models*), 583
imgparam (*attribut sirilpy.models.Sequence*), 589
import_onnxruntime() (*méthode sirilpy.gpuhelper.ONNXHelper*), 601
imul, **498**
incl (*attribut sirilpy.models.ImgData*), 583
index (*attribut sirilpy.models.DistoData*), 575
info_messagebox() (*méthode sirilpy.connection.SirilInterface*), 565
Inliers (*attribut sirilpy.models.Homography*), 580
inspector, **498**
install_jax() (*méthode sirilpy.gpuhelper.JaxHelper*), 599
install_onnxruntime() (*méthode sirilpy.gpuhelper.ONNXHelper*), 601
install_torch() (*méthode sirilpy.gpuhelper.TorchHelper*), 602
instrume (*attribut sirilpy.models.FKeywords*), 579
invght, **499**
invmodasinh, **499**
invmtf, **499**
is_cli() (*méthode sirilpy.connection.SirilInterface*), 565
is_image_loaded() (*méthode sirilpy.connection.SirilInterface*), 566
is_jax_installed() (*méthode sirilpy.gpuhelper.JaxHelper*), 599
is_onnxruntime_installed() (*méthode sirilpy.gpuhelper.ONNXHelper*), 601
is_sequence_loaded() (*méthode sirilpy.connection.SirilInterface*), 566
is_torch_installed() (*méthode sirilpy.gpuhelper.TorchHelper*), 603
is_variable (*attribut sirilpy.models.Sequence*), 589
iso_speed (*attribut sirilpy.models.FKeywords*), 579
isub, **500**

J

JaxHelper (*classe dans sirilpy.gpuhelper*), 599

jsonmetadata, **500**

K

keywords (*attribut sirilpy.models.FFit*), 577
KEYWORDS_DIALOG (*attribut sirilpy.enums.DialogID*), 591
KEYWORDS_FORMAT (*attribut sirilpy.models.FKeywords*), 578
KEYWORDS_SIZE (*attribut sirilpy.models.FKeywords*), 578

L

layer (*attribut sirilpy.models.PSFStar*), 585
legend (*attribut sirilpy.models.Polygon*), 587
LIGHT (*attribut sirilpy.enums.ImageType*), 592
light_curve, **500**
limit, **501**
LINEAR_DISPLAY (*attribut sirilpy.enums.STFType*), 593
linear_match, **501**
LINEARMATCH_DIALOG (*attribut sirilpy.enums.DialogID*), 591
LINES (*attribut sirilpy.enums.PlotType*), 593
LINESHYPHENS (*attribut sirilpy.enums.PlotType*), 593
LINESMARKS (*attribut sirilpy.enums.PlotType*), 593
LINESPOINTS (*attribut sirilpy.enums.PlotType*), 593
link, **502**
linstretch, **502**
livestack, **502**
livetime (*attribut sirilpy.models.FKeywords*), 579
lo (*attribut sirilpy.models.FKeywords*), 579
load, **503**
load_image_from_file() (*méthode sirilpy.connection.SirilInterface*), 566
load_seq, **503**
location (*attribut sirilpy.models.ImageStats*), 582
log, **503**
log() (*méthode sirilpy.connection.SirilInterface*), 566
LOG_DISPLAY (*attribut sirilpy.enums.STFType*), 593
LogColor (*classe dans sirilpy.enums*), 593
LONG_IMG (*attribut sirilpy.enums.BitpixType*), 589
ls, **503**

M

mad (*attribut sirilpy.models.ImageStats*), 582
mag (*attribut sirilpy.models.PSFStar*), 585
makepsf, **504**
MARKS (*attribut sirilpy.enums.PlotType*), 593
match_theme_to_siril() (*dans le module sirilpy.tksiril*), 605
max (*attribut sirilpy.models.BGSample*), 575
max (*attribut sirilpy.models.ImageStats*), 582
maxi (*attribut sirilpy.models.FFit*), 577
mean (*attribut sirilpy.models.BGSample*), 575
mean (*attribut sirilpy.models.ImageStats*), 582
median (*attribut sirilpy.models.BGSample*), 575
median (*attribut sirilpy.models.ImageStats*), 582
MEDIAN_DIALOG (*attribut sirilpy.enums.DialogID*), 591
merge, **505**
merge_cfa, **505**
MERGE_CFA_DIALOG (*attribut sirilpy.enums.DialogID*), 591
min (*attribut sirilpy.models.BGSample*), 575
min (*attribut sirilpy.models.ImageStats*), 582
mini (*attribut sirilpy.models.FFit*), 577
MINMAX (*attribut sirilpy.enums.SlidersMode*), 594
MIPSLOHI (*attribut sirilpy.enums.SlidersMode*), 594
mirrorx, **505**
mirrorx_single, **506**
mirrory, **506**
modasinh, **506**
module
    sirilpy.connection, 553
    sirilpy.enums, 589
    sirilpy.exceptions, 606
    sirilpy.gpuhelper, 599
    sirilpy.models, 574
    sirilpy.plot, 594
    sirilpy.tksiril, 604
    sirilpy.utility, 595
MOFFAT (*attribut sirilpy.enums.StarProfile*), 594
MOFFAT_FIXED (*attribut sirilpy.enums.StarProfile*), 594
MONO (*attribut sirilpy.enums.DialogReq*), 592
MONO (*attribut sirilpy.enums.SirilVport*), 594
MouseModeError, 606
mtf, **507**

N

naxes (*propriété sirilpy.models.FFit*), 577
naxis (*attribut sirilpy.models.FFit*), 577
nb_layers (*attribut sirilpy.models.Sequence*), 589
nbstars (*attribut sirilpy.models.ImageAnalysis*), 582
needs_module_version() (*dans le module sirilpy.utility*), 597
neg, **507**
neg_ratio (*attribut sirilpy.models.FFit*), 577
new, **507**
ngoodpix (*attribut sirilpy.models.ImageStats*), 582
NINA_LIGHT_CURVE (*attribut sirilpy.enums.DialogID*), 591
NoImageError, 606
NONE (*attribut sirilpy.enums.DialogReq*), 592
normValue (*attribut sirilpy.models.ImageStats*), 582
NoSequenceError, 607
nozero, **508**
number (*attribut sirilpy.models.Sequence*), 589
number_of_stars (*attribut sirilpy.models.RegData*), 588

O

objctdec (*attribut sirilpy.models.FKeywords*), 579
objctra (*attribut sirilpy.models.FKeywords*), 579
object (*attribut sirilpy.models.FKeywords*), 579
observer (*attribut sirilpy.models.FKeywords*), 580
offline, **508**
offset, **508**
offset (*attribut sirilpy.models.FKeywords*), 580
online, **508**
ONNXHelper (*classe dans sirilpy.gpuhelper*), 600
OPEN_DIALOG (*attribut sirilpy.enums.DialogID*), 591
open_dialog() (*méthode sirilpy.connection.SirilInterface*), 567
orig_bitpix (*attribut sirilpy.models.FFit*), 577
overlay_add_polygon() (*méthode sirilpy.connection.SirilInterface*), 567
overlay_clear_polygons() (*méthode sirilpy.connection.SirilInterface*), 567
overlay_delete_polygon() (*méthode sirilpy.connection.SirilInterface*), 567
overlay_draw_polygon() (*méthode sirilpy.connection.SirilInterface*), 567
overlay_get_polygon() (*méthode sirilpy.connection.SirilInterface*), 568
overlay_get_polygons_list() (*méthode sirilpy.connection.SirilInterface*), 568

P

pair_matched (*attribut sirilpy.models.Homography*), 581
parse, **509**
parse_fits_header() (*dans le module sirilpy.utility*), 597
pcc, **509**
PCC_DIALOG (*attribut sirilpy.enums.DialogID*), 591
phot_is_valid (*attribut sirilpy.models.PSFStar*), 585
pix2radec() (*méthode sirilpy.connection.SirilInterface*), 568
PIXEL_MATH_DIALOG (*attribut sirilpy.enums.DialogID*), 591
pixel_size_x (*attribut sirilpy.models.FKeywords*), 580
pixel_size_y (*attribut sirilpy.models.FKeywords*), 580
platesolve, **509**
PlotData (*classe dans sirilpy.plot*), 594
PlotType (*classe dans sirilpy.enums*), 593
PLTSOLVD (*attribut sirilpy.enums.DialogReq*), 592
pltsolvd (*attribut sirilpy.models.FKeywords*), 580
pltsolvd_comment (*attribut sirilpy.models.FKeywords*), 580
pm, **511**
POINTS (*attribut sirilpy.enums.PlotType*), 593
points (*attribut sirilpy.models.Polygon*), 587
Polygon (*classe dans sirilpy.models*), 585
polygon_id (*attribut sirilpy.models.Polygon*), 587
position (*attribut sirilpy.models.BGSample*), 575
PREFS_DIALOG (*attribut sirilpy.enums.DialogID*), 591
ProcessingThreadBusyError, 607
profile, **511**
profile (*attribut sirilpy.models.PSFStar*), 585
program (*attribut sirilpy.models.FKeywords*), 580
psf, **511**
PSFStar (*classe dans sirilpy.models*), 583
pwd, **512**
pyscript, **512**

Q

quality (*attribut sirilpy.models.RegData*), 588

R

R (*attribut sirilpy.models.PSFStar*), 584
ra (*attribut sirilpy.models.FKeywords*), 580
ra (*attribut sirilpy.models.PSFStar*), 585
radec2pix() (*méthode sirilpy.connection.SirilInterface*), 568
RED (*attribut sirilpy.enums.LogColor*), 593
RED (*attribut sirilpy.enums.SirilVport*), 594
redo() (*méthode sirilpy.connection.SirilInterface*), 569
reference_image (*attribut sirilpy.models.Sequence*), 589
RegData (*classe dans sirilpy.models*), 587
register, **512**
regparam (*attribut sirilpy.models.Sequence*), 589
reloadscripts, **514**
requires, **514**
resample, **514**
RESAMPLE_DIALOG (*attribut sirilpy.enums.DialogID*), 591
reset_progress() (*méthode sirilpy.connection.SirilInterface*), 569
RGB (*attribut sirilpy.enums.DialogReq*), 592
RGB (*attribut sirilpy.enums.SirilVport*), 594
rgbcomp, **515**
RGBPLTSOLVD (*attribut sirilpy.enums.DialogReq*), 592
rgradient, **515**
RGRADIENT_DIALOG (*attribut sirilpy.enums.DialogID*), 591
rl, **515**
rmgreen, **516**
rmse (*attribut sirilpy.models.PSFStar*), 585
rotate, **516**
rotatePi, **516**
ROTATION_DIALOG (*attribut sirilpy.enums.DialogID*), 591
roundness (*attribut sirilpy.models.ImageAnalysis*), 582
roundness (*attribut sirilpy.models.RegData*), 588
row_order (*attribut sirilpy.models.FKeywords*), 580
run() (*méthode sirilpy.gpuhelper.ONNXHelper*), 601
rx (*attribut sirilpy.models.ImgData*), 583

rx (*attribut sirilpy.models.Sequence*), 589
ry (*attribut sirilpy.models.ImgData*), 583
ry (*attribut sirilpy.models.Sequence*), 589

S

s_Bmag (*attribut sirilpy.models.PSFStar*), 585
s_mag (*attribut sirilpy.models.PSFStar*), 585
S_PCC_DIALOG (*attribut sirilpy.enums.DialogID*), 591
safe_subprocess_Popen() (*dans le module sirilpy.utility*), 598
safe_subprocess_run() (*dans le module sirilpy.utility*), 598
SALMON (*attribut sirilpy.enums.LogColor*), 593
sat (*attribut sirilpy.models.PSFStar*), 585
satu, **517**
SATU_DIALOG (*attribut sirilpy.enums.DialogID*), 591
save, **517**
save_image_file() (*méthode sirilpy.connection.SirilInterface*), 569
SAVEAS_DIALOG (*attribut sirilpy.enums.DialogID*), 591
savebmp, **517**
savejpg, **518**
savejxl, **518**
savepng, **518**
savepnm, **519**
savetif, **519**
savetif32, **519**
savetif8, **519**
sb, **520**
scale (*attribut sirilpy.models.ImageStats*), 583
SCNR_DIALOG (*attribut sirilpy.enums.DialogID*), 591
ScrollableFrame (*classe dans sirilpy.tksiril*), 604
select, **520**
selnum (*attribut sirilpy.models.Sequence*), 589
SEQ (*attribut sirilpy.enums.DialogReq*), 592
SEQ_AVI (*attribut sirilpy.enums.SequenceType*), 593
SEQ_FITSEQ (*attribut sirilpy.enums.SequenceType*), 593
SEQ_INTERNAL (*attribut sirilpy.enums.SequenceType*), 593
SEQ_REGULAR (*attribut sirilpy.enums.SequenceType*), 593
SEQ_SER (*attribut sirilpy.enums.SequenceType*), 593
seqapplyreg, **520**
seqccm, **522**
seqclean, **522**
seqcosme, **523**
seqcosme_cfa, **523**
seqcrop, **523**
seqextract_Green, **524**
seqextract_Ha, **524**
seqextract_HaOIII, **524**
seqfind_cosme, **524**
seqfind_cosme_cfa, **525**
seqfindstar, **525**
seqfixbanding, **525**
seqght, **526**
seqheader, **526**
seqinvght, **526**
seqinvmodasinh, **526**
seqlinstretch, **527**
SEQLIST_DIALOG (*attribut sirilpy.enums.DialogID*), 591
seqmerge_cfa, **527**
seqmodasinh, **527**
seqmtf, **528**
seqname (*attribut sirilpy.models.Sequence*), 589
seqplatesolve, **529**
seqprofile, **528**
seqpsf, **528**
seqresample, **530**
seqrl, **530**
seqsb, **531**
seqsetmag, **531**
seqsplit_cfa, **531**
seqstarnet, **532**
seqstat, **532**
seqsubsky, **533**
seqtilt, **533**
Sequence (*classe dans sirilpy.models*), 588
SequenceType (*classe dans sirilpy.enums*), 593
sequnsetmag, **533**
sequpdate_key, **533**
seqwiener, **534**
serialize() (*méthode de la classe sirilpy.plot.PlotData*), 594
serialize() (*méthode sirilpy.models.ImageAnalysis*), 581
serialize() (*méthode sirilpy.models.Polygon*), 587
SeriesData (*classe dans sirilpy.plot*), 595
set, **534**
set16bits, **534**
set32bits, **535**
set_image_bgsamples() (*méthode sirilpy.connection.SirilInterface*), 569
set_image_filename() (*méthode sirilpy.connection.SirilInterface*), 570
set_image_iccprofile() (*méthode sirilpy.connection.SirilInterface*), 570
set_image_metadata_from_header_string() (*méthode sirilpy.connection.SirilInterface*), 570
set_image_pixeldata() (*méthode sirilpy.connection.SirilInterface*), 571
set_seq_frame_incl() (*méthode sirilpy.connection.SirilInterface*), 571
set_seq_frame_pixeldata() (*méthode sirilpy.connection.SirilInterface*), 571
set_siril_pan() (*méthode sirilpy.connection.SirilInterface*), 572
set_siril_selection() (*méthode sirilpy.connection.SirilInterface*), 572

set_siril_slider_lohi() (*méthode sirilpy.connection.SirilInterface*), 572
set_siril_slider_mode() (*méthode sirilpy.connection.SirilInterface*), 572
set_siril_stf() (*méthode sirilpy.connection.SirilInterface*), 573
set_siril_stf_linked() (*méthode sirilpy.connection.SirilInterface*), 573
set_siril_zoom() (*méthode sirilpy.connection.SirilInterface*), 573
set_temp (*attribut sirilpy.models.FKeywords*), 580
setcompress, **535**
setcpu, **535**
setext, **535**
setfindstar, **536**
setmag, **537**
setmem, **537**
setphot, **537**
setref, **538**
SharedMemoryError, 607
SHORT_IMG (*attribut sirilpy.enums.BitpixType*), 589
show, **538**
sigma (*attribut sirilpy.models.ImageStats*), 583
SirilConnectionError, 607
SirilError, 607
SirilInterface (*classe dans sirilpy.connection*), 553
sirilpy.connection
    module, 553
sirilpy.enums
    module, 589
sirilpy.exceptions
    module, 606
sirilpy.gpuhelper
    module, 599
sirilpy.models
    module, 574
sirilpy.plot
    module, 594
sirilpy.tksiril
    module, 604
sirilpy.utility
    module, 595
SirilVport (*classe dans sirilpy.enums*), 593
siteelev (*attribut sirilpy.models.FKeywords*), 580
sitelat (*attribut sirilpy.models.FKeywords*), 580
sitelat_str (*attribut sirilpy.models.FKeywords*), 580
sitelong (*attribut sirilpy.models.FKeywords*), 580
sitelong_str (*attribut sirilpy.models.FKeywords*), 580
size (*attribut sirilpy.models.BGSample*), 575
SlidersMode (*classe dans sirilpy.enums*), 594
SNR (*attribut sirilpy.models.PSFStar*), 584
spcc, **538**
spcc_list, **539**
split, **540**
split_cfa, **540**
SPLIT_CFA_DIALOG (*attribut sirilpy.enums.DialogID*), 591
SQRT_DISPLAY (*attribut sirilpy.enums.STFType*), 593
sqrtbwmv (*attribut sirilpy.models.ImageStats*), 583
SQUARED_DISPLAY (*attribut sirilpy.enums.STFType*), 593
stack, **540**
stackall, **542**
stackcnt (*attribut sirilpy.models.FKeywords*), 580
standard_style() (*dans le module sirilpy.tksiril*), 605
star_name (*attribut sirilpy.models.PSFStar*), 585
starnet, **542**
STARNET_DIALOG (*attribut sirilpy.enums.DialogID*), 591
StarProfile (*classe dans sirilpy.enums*), 594
STARS_LIST_WINDOW (*attribut sirilpy.enums.DialogID*), 591
start_ls, **543**
stat, **543**
STAT_WINDOW (*attribut sirilpy.enums.DialogID*), 591
stats (*attribut sirilpy.models.FFit*), 577
stats (*attribut sirilpy.models.Sequence*), 589
status() (*méthode sirilpy.gpuhelper.JaxHelper*), 599
status() (*méthode sirilpy.gpuhelper.ONNXHelper*), 601
status() (*méthode sirilpy.gpuhelper.TorchHelper*), 603
status_code (*attribut sirilpy.exceptions.CommandError*), 606
STFType (*classe dans sirilpy.enums*), 593
stop_ls, **543**
subsky, **544**
SuppressedStderr (*classe dans sirilpy.utility*), 595
SuppressedStdout (*classe dans sirilpy.utility*), 595
sx (*attribut sirilpy.models.PSFStar*), 585
sx_err (*attribut sirilpy.models.PSFStar*), 585
sy (*attribut sirilpy.models.PSFStar*), 585
sy_err (*attribut sirilpy.models.PSFStar*), 585
synthstar, **544**

T

telescop (*attribut sirilpy.models.FKeywords*), 580
test_jax() (*méthode sirilpy.gpuhelper.JaxHelper*), 599
test_onnxruntime() (*méthode sirilpy.gpuhelper.ONNXHelper*), 602
test_torch() (*méthode sirilpy.gpuhelper.TorchHelper*), 603
thresh, **545**
threshhi, **545**
threshlo, **545**
tilt, **545**
timestamp (*attribut sirilpy.models.ImageAnalysis*), 582
top_down (*attribut sirilpy.models.FFit*), 577
TorchHelper (*classe dans sirilpy.gpuhelper*), 602
total (*attribut sirilpy.models.ImageStats*), 583
trixel, **546**

truncate_utf8() (*dans le module sirilpy.utility*), 598
type (*attribut sirilpy.models.Sequence*), 589

U

unclipstars, **546**
undo() (*méthode sirilpy.connection.SirilInterface*), 573
undo_save_state() (*méthode sirilpy.connection.SirilInterface*), 573
uninstall_jax() (*méthode sirilpy.gpuhelper.JaxHelper*), 600
uninstall_onnxruntime() (*méthode sirilpy.gpuhelper.ONNXHelper*), 602
uninstall_package() (*dans le module sirilpy.utility*), 598
uninstall_torch() (*méthode sirilpy.gpuhelper.TorchHelper*), 603
units (*attribut sirilpy.models.PSFStar*), 585
UNKNOWN (*attribut sirilpy.enums.ImageType*), 592
unknown_keys (*attribut sirilpy.models.FFit*), 577
unpurple, **546**
UNPURPLE_DIALOG (*attribut sirilpy.enums.DialogID*), 591
unselect, **547**
unsetmag, **547**
unsharp, **547**
update_key, **547**
update_progress() (*méthode sirilpy.connection.SirilInterface*), 573
update_stats() (*méthode sirilpy.models.FFit*), 576
USER (*attribut sirilpy.enums.SlidersMode*), 594
USHORT_IMG (*attribut sirilpy.enums.BitpixType*), 589

V

valid (*attribut sirilpy.models.BGSample*), 575
velocity (*attribut sirilpy.models.DistoData*), 575
visu, **548**

W

warning_messagebox() (*méthode sirilpy.connection.SirilInterface*), 574
wavelet, **548**
WAVELETS_DIALOG (*attribut sirilpy.enums.DialogID*), 592
weighted_fwhm (*attribut sirilpy.models.RegData*), 588
wfwhm (*attribut sirilpy.models.ImageAnalysis*), 582
width (*attribut sirilpy.models.ImageAnalysis*), 582
width (*propriété sirilpy.models.FFit*), 577
wiener, **548**
wrecons, **549**

X

x (*attribut sirilpy.models.FPoint*), 580
x0 (*attribut sirilpy.models.PSFStar*), 585
x_err (*attribut sirilpy.models.PSFStar*), 585
xpos (*attribut sirilpy.models.PSFStar*), 585
xy_plot() (*méthode sirilpy.connection.SirilInterface*), 574

Y

y (*attribut sirilpy.models.FPoint*), 580
y0 (*attribut sirilpy.models.PSFStar*), 585
y_err (*attribut sirilpy.models.PSFStar*), 585
ypos (*attribut sirilpy.models.PSFStar*), 585