Skip to content, Skip to search

This page contains a list of commands available in Siril 1.2. The reference for version 1.0 is available here (https://free-astro.org/index.php?title=Siril:Commands&oldid=8045).

Commands are often added or modified, this means that the version of Siril you are using may not be the same as the version documented here.

Obsah

Introduction

Siril has a command line in its graphical user interface and an ability to run scripts that are a list of commands, either from the graphical user interface or from the command line interface. In general, commands that modify a single image work on the currently loaded image, so the use of the LOAD command is required in scripts, and commands that work on a sequence of images take the name of the sequence as argument. If files are not named in a way that Siril detects as a sequence, the command CONVERT will help. If the currently loaded sequence (from GUI) is to be used, the special name **.** can be used instead of the full sequence name.

The `<SPACE>` character is the delimiter between arguments. If you need to have spaces inside the arguments, you can use the quote or double quote, just like in a shell.

Commands can be typed in the command line at the bottom of Siril's main window. Another way is to put commands in a file and execute it as a script. To execute the file from the GUI, add it to the configured script directories or from the GUI, use the @ token of the command line like so:

```
@file_name
```

Some commands (preprocess, stack, and all save commands) can use file names containing variables coming from the FITS header. The format of the expression is explained in details here (https://gitlab.com/free-astro/siril/-/merge_requests/346) and can be tested using the parse command.

Command stream (pipe)

Starting with 0.9.10, a new mode has been introduced in which commands can be sent through a named pipe and logs and status can be obtained through another. The mode is activated using the `-p` command line argument.

The protocol is quite simple: Siril receives commands and outputs some updates. Only commands that are marked as scriptable can be used with this system. Whenever the command inputs pipe is closed or the `cancel` command is given, the running command is stopped as if the stop button was clicked on in the GUI. The pipes are named `siril_command.in` and `siril_command.out` and are available in /tmp on Unix-based systems. Since version 1.2.0, the paths of the pipes can be configured with `-r` and `-w` options, which allows external programs to create them before starting Siril, typically with the `mkfifo` command. Also new in 1.2.0, a `ping` command will simply give a status return, indicating if siril is ready to process a command or already busy.

Outputs of siril on the pipe is a stream of one line text and formatted as follows:

- `ready` is printed on startup, indicating siril is ready to process commands
- `log:` followed by a log message
- `status: verb [subject]`, where verb can be either of `starting`, `success`, `error` or `exit` (exit message is not yet implemented). The subject is the current command name, except for exit that indicates that siril suffered a fatal error and has to exit.
- `progress: value%` is the equivalent of the progress bar, it sends percents periodically, and sometimes 0% at the end of a processing as an *idle* information.

Commands history

These lists help you see what has changed in the last version, whether new or modified commands.

- 1.2.0: **update of** stack (changing weights options, adding rgb equalization, adding possibility to disable rejection, adding the fastnorm option to provide a faster alternative to normalization computation, adding k.sigma frame filters, adding rejection maps), seqstat (adding full option), register (adding maxstars, 2pass, noout and interp options), preprocess (adding cosmetic correction options), setfindstar (adding all options and relaxed mode), mirrorx (adding bottomup option), adding the CFA and main option to stat and seqstat, remade rl with the new deconvolution, reviewing doc and arguments of commands satu, rmgreen, asinh, boxselect, psf, seqpsf subsky, seqsubsky, help, neg.
  New commands preprocess_single, autostretch, seqtilt, dumpheader, setphot, start_ls, livestack, stop_ls, rgbcomp, seqapplyreg, seqclean, pcc, pm, ght, invght, invmodasinh modasinh, linstretch, nomad, get, set, starnet, light_curve, denoise, invmtf, jsonmetadata, parse, binxy, mirrorx_single merge_cfa, seqheader, seqfind_cosme_cfa, seqfindstar, seqfixbanding, seqmerge_cfa, synthstar, unclipstars, capabilities, inspector, makepsf, sb, wiener, seqrl, seqsb, seqwiener, autoghs, platesolve, seqplatesolve, getref, calibrate, calibrate_single, seqsetmag, sequnsetmag, seqstarnet, show, seqght, seqinvght, seqinvmodasinh, seqlinstretch, seqmodasinh
- 1.0.0: **update of** rl, stack, stackall (adding -output_norm and rejection algorithms), preprocess (removing -stretch and -flip and adding -fix_xtrans) and convertraw (adding -start=index, -out=directory and -fitseq -ser). Also, a `-prefix=` option has been added to the sequence commands that build a new sequence.
  New commands subsky, seqsubsky, neg, mtf, seqmtf, linear_match, extract_Ha, extract_HaOIII, seqextract_Ha, set16bits, set32bits, setcompress, seqextract_HaOIII, link, convert, reloadscripts, fix_xtrans, requires, merge, seqstat, setref, seqextract_Green, extract_Green, tilt, boxselect
- 0.9.12: split_cfa, seqsplit_cfa
- 0.9.11: stack, stackall, setmem
- 0.9.10: setfindstar, grey_flat, asinh, rgradient
- 0.9.9: convertraw, preprocess, register, stack, close, clear, setext, @
- 0.9.8: savepng, stackall,

Siril command line functions reference

This is the list of commands available in the development version of Siril. Check the list above, some may not be available in your version.

addmax

```
addmax filename
```

addmax compute a new image IMG with IMG_1 and IMG_2. The pixel of IMG_1 is replaced by the pixel at the same coordinates of IMG_2 if the intensity of 2 is greater than 1. Do not forget to save the result.

asinh

```
asinh [-human] stretch [offset]
```

Stretches the image to show faint objects using an hyperbolic arcsin transformation.

The mandatory argument `stretch`, typically between 1 and 1000, will give the strength of the stretch. The black point can be offset by providing an optional `offset` argument in the normalized pixel value of [0, 1]. Finally the option `-human` uses human eye luminous efficiency weights to compute the luminance used to compute the stretch value for each pixel, instead of the simple mean of the channels pixel values.

This stretch method preserves lightness from the L*a*b* color space.

autoghs

```
autoghs [-linked] shadowsclip stretchamount [-b=] [-hp=] [-lp=]
```

Application of the generalized hyperbolic stretch with a symmetry point SP defined as k.sigma from the median of each channel (the provided `shadowsclip` value is the k here and can be negative).

By default, SP and the stretch are computed per channel; SP can be computed as a mean of image channels by passing `-linked`.

The stretch amount `D` is provided in the second mandatory argument. Implicit values of 13 for `B`, making it very focused on the SP brightness range, 0.7 for `HP`, 0 for `LP` are used but can be changed with the options of the same names.

autostretch

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

Auto-stretches the currently loaded image, with different parameters for each channel (unlinked) unless `-linked` is passed. Arguments are optional, `shadowclip` is the shadows clipping point, measured in sigma units from the main histogram peak (default is -2.8), `targetbg` is the target background value, giving a final brightness to the image, range [0, 1], default is 0.25. The default values are those used in the Auto-stretch rendering from the GUI.

Do not use the unlinked version after color calibration. Linked preserves the white balance.

bg

```
bg
```

Returns the background level of the image loaded in memory.

bgnoise

```
bgnoise
```

Returns the background noise level.

binxy

```
binxy coefficient [-sum]
```

Computes the numerical binning of the in-memory image (sum of the pixels 2x2, 3x3..., like the analogic binning of CCD camera). If the optional argument `-sum` is passed, then the sum of pixels is computed, while it is the average when no optional argument is provided.

boxselect

```
boxselect [-clear] [x y width height]
```

Make a selection area in the currently loaded image with the arguments `x`, `y`, `width` and `height`, with `x` and `y` being the coordinates of the top left corner starting at (0, 0), and `width` and `height` the size of the selection.

The `-clear` argument deletes any selection area. If no argument is passed, the current selection is printed.

calibrate

```
calibrate sequencename [-bias=filename|value] [-dark=filename] [-flat=filename] [{ -cc=dark [siglo sighi] | -cc=bpm
bpmfile }] [-cfa] [-debayer] [-fix_xtrans] [-equalize_cfa] [-opt] [-prefix=] [-fitseq]
```

Calibrates the sequence `sequencename` using any of bias, dark and flat masters given in argument.

The bias master file can be replaced by an integer value or an expression that uses the `OFFSET` entry from the FITS header, in the form `=2048` or `=64*$OFFSET`. See the tutorial on synthetic bias (https://siril.org/tutorials/synthetic-biases/#and-now-what) for more information.

By default, cosmetic correction is not activated. If you wish to apply some, you will need to specify it with `-cc=` option. You can use `-cc=dark` to detect hot and cold pixels from the masterdark (a masterdark must be given with the `-dark=` option), optionally followed by `siglo` and `sighi` for cold and hot pixels respectively. A value of 0 deactivates the correction. If sigmas are not provided, only hot pixels detection with a sigma of 3 will be applied. Alternatively, you can use `-cc=bpm` followed by the path to your Bad Pixel Map to specify which pixels must be corrected. An example file can be obtained with a FIND_HOT command on a masterdark.

Three options apply to color images (in CFA format): `-cfa` for cosmetic correction purposes, `-debayer` to demosaic them before saving and `-equalize_cfa` to equalize the mean intensity of the three R, G and B layers of the CFA flat master to avoid giving a tint to the calibrated image.

The `-fix_xtrans` option is dedicated to X-Trans files by applying a correction on dark and bias images to remove an ugly square pattern caused by autofocus.

It is also possible to optimize (adjust its level to match the offset of the image) the dark subtraction with `-opt`, possible only if both bias and dark masters are provided.

The output sequence name starts with the prefix "pp_" unless otherwise specified with option `-prefix=`. If `-fitseq` is provided, the output sequence will be a FITS sequence (single file).

calibrate_single

```
calibrate_single filename [-bias=filename|value] [-dark=filename] [-flat=filename] [{ -cc=dark [siglo sighi] | -cc=bpm
bpmfile }] [-cfa] [-debayer] [-fix_xtrans] [-equalize_cfa] [-opt] [-prefix=]
```

Calibrates a single FITS image, using any of bias, dark and flat masters given in argument. Arguments are the same as for CALIBRATE, except `-fitseq` that applies only to a sequence.

capabilities

```
capabilities
```

Lists Siril capabilities, based on compilation and runtime.

cd

```
cd directory
```

Set the new current working directory. `directory` can contain the ~ token, expanded as the home directory, directories with spaces in the name can be protected using single or double quotes. Examples:

- `cd ~/M42`
- `cd '../OIII 2x2/'`

cdg

```
cdg
```

Return the coordinates of the center of gravity of the image. Only pixels with values above 15.7% of max ADU and having four neighbors filling the same condition are used to compute it, and it is computed only if there are at least 50 of them.

clahe

```
clahe cliplimit tileSize
```

Equalizes the histogram of an image using Contrast Limited Adaptive Histogram Equalization

clear

```
clear
```

Clears the graphical output logs

clearstar

```
clearstar
```

Clear all the stars saved in memory and displayed on the screen.

close

```
close
```

Properly closes the opened image and the opened sequence, if any.

convert

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

Convert all images of the current working directory that are in a supported format into Siril's sequence of FITS images (several files) or a FITS sequence (single file) if `-fitseq` is provided or a SER sequence (single file) if `-ser` is provided. The argument `basename` is the base name of the new sequence, numbers and the extension will be put behind it.

For FITS images, Siril will try to make a symbolic link; if not possible, files will be copied. The option `-debayer` demosaics the input CFA images, in this case new RGB files are created.

`-start=index` sets the starting index parameter for the converted images (not used with `-fitseq` or `-ser`) useful to continue an existing sequence (make sure you remove the target .seq if it exists in that case, or that you check *force .seq recomputation* when searching sequences.

The `-out=` option changes the output directory to the provided argument.

convertraw

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

Convert DSLR RAW files into Siril's FITS images or a FITS sequence (single file / FITS cube) if `-fitseq` is provided or a SER sequence (single file) if `-ser` is provided. The basename argument is the base name of the new sequence. The `debayer` option applies demosaicing to images, `-start=index` sets the starting index parameter and `-out=directory` provides a directory for output files inseead of the CWD. For more information on arguments see convert.

cosme

```
cosme filename
```

Apply the local mean to a set of pixels on the in-memory image (cosmetic correction). The coordinate of this pixels are in an ASCII file (provided in the `filename` argument) in lines whose format is indicated below. COSME is adapted to correct residual hot and cold pixels after preprocessing.

Lines `P x y type` will fix the pixel at coordinates `(x, y)` `type` is an optional character (`C` or `H`) specifying to Siril if the current pixel is cold or hot. This line is created by the command FIND_HOT but you also can add some lines manually:
Lines `C x 0 type` will fix the bad column at coordinates `x`.
Lines `L y 0 type` will fix the bad line at coordinates `y`.

Instead of providing the list of bad pixels, it's also possible to detect them in the current image using the FIND_COSME command.

cosme_cfa

```
cosme_cfa filename
```

Same function as COSME but applying to RAW CFA images.

crop

```
crop [x, y, width, height]
```

It can be used with the GUI: if a selection has been made with the mouse, calling the crop command without arguments crops it on this selection. Otherwise, or in scripts, arguments have to be given, with x and y being the coordinates of the top left corner, and width and height the size of the selection.

ddp

```
ddp level coef sigma
```

Performs a DDP (digital development processing) as described first by Kunihiko Okano. This implementation is the one described in IRIS (http://www.astrosurf.org/buil/us/iris/iris5.htm). It combines a linear distribution on low levels (below level) and a non-linear on high levels. It uses a Gaussian filter of sigma sigma multiplies the resulting image by coef. The typical values for sigma are included between 0.7 and 2

denoise

```
denoise [-nocosmetic] [-mod=m] [ -vst | -da3d | -sos=n [-rho=r] ] [-indep]
```

Denoises the image using the non-local Bayesian algorithm described by Lebrun, Buades and Morel in 2013 (https://www.ipol.im/pub/art/2013/16).

It is strongly recommended to apply cosmetic correction to remove salt and pepper noise before running denoise, and by default this command will apply cosmetic correction automatically. However, if this has already been carried out earlier in the workflow it may be disabled here using the optional command -`nocosmetic`.

An optional parameter `-mod=m` may be given, where 0 <= m <= 1. The output pixel is computed as : $out = m \times d + (1 - m) \times in$, where *d* is the denoised pixel value. A modulation value of 1 will apply no modulation. If the parameter is omitted, it defaults to 1.

The optional parameter `-vst` can be used to apply the generalised Anscombe variance stabilising transform prior to NL-Bayes. This is useful with photon-starved images such as single subs, where the noise follows a Poisson or Poisson-Gaussian distribution rather than being primarily Gaussian. It cannot be used in conjunction with DA3D or SOS, and for denoising stacked images it is usually not beneficial.

The optional parameter `-da3d` can be used to enable Data-Adaptive Dual Domain Denoising (DA3D) as a final stage denoising algorithm. This uses the output of BM3D as a guide image to refine the denoising. It improves detail and reduces staircasing artefacts.

The optional parameter `-sos=n` can be used to enable Strengthen-Operate-Subtract (SOS) iterative denoise boosting, with the number of iterations specified by n. In particular, this booster may produce better results if the un-boosted NL-Bayes algorithm produces artefacts in background areas. If both -da3d and -sos=n are specified, the last to be specified will apply.

The optional parameter `-rho=r` may be specified, where 0 < r < 1. This is used by the SOS booster to determine the amount of noisy image added in to the intermediate result between each iteration. If `-sos=` is not specified then this parameter is ignored.

The default is not to apply DA3D or SOS, as the improvement in denoising is usually relatively small and these techniques requires additional processing time.

In very rare cases, blocky coloured artefacts may be found in the output when denoising colour images. The optional argument `-indep` can be used to prevent this by denoising each channel separately. This is slower but will eliminate artefacts

dumpheader

```
dumpheader
```

Prints the FITS header of the currently loaded image, if any.

entropy

```
entropy
```

Computes the entropy of the opened image on the displayed layer, only in the selected area if one has been selected or in the whole image else. The entropy is one way of measuring the noise or the details in an image.

exit

```
exit
```

Quits the application.

extract

```
extract NbPlane
```

Extracts NbPlane planes of Wavelet domain. For color extraction, see split.

extract_Green

```
extract_Green
```

Extracts green signal from the currently loaded CFA image. It reads the Bayer matrix information from the image or the preferences and exports only the averaged green filter data as a new half-sized FITS file. The output file name starts with the prefix "Green_".

See SEQEXTRACT_GREEN for the same operation for sequences.

extract_Ha

```
extract_Ha
```

Extracts H-alpha signal from the currently loaded CFA image. It reads the Bayer matrix information from the image or the preferences and exports only the red filter data as a new half-sized FITS file. The output file name starts with the prefix "Ha_".

See SEQEXTRACT_HA for the same operation for sequences.

extract_HaOIII

```
extract_HaOIII
```

Extracts H-alpha and O-III signals from the currently loaded CFA image. It reads the Bayer matrix information from the image or the preferences and exports only the red filter data for H-alpha and an average of the three others as a new half-sized FITS files. The output file name start with the prefixes "Ha_" and "OIII_".

See SEQEXTRACT_HAOII for the same operation for sequences.

fdiv

```
fdiv filename scalar
```

Divides the image in memory by the image given in argument. The resulting image is multiplied by the value of the scalar argument. Please check that the image is in the working directory.

See also idiv.

ffill

```
ffill value x y width height
```

Same command as FILL but this is a symmetric fill of a region defined by the mouse. Used to process an image in the Fourier (FFT) domain.

fftd

```
fftd modulus phase
```

Applies a Fast Fourier Transform to the image loaded in memory. Modulus and phase given in argument are saved in FITS files.

ffti

```
ffti modulus phase
```

This function is used to retrieve corrected image applying an inverse transformation. The modulus and phase used are the files given in argument.

fill

```
fill value x y width height
```

Fills the whole current image (or selection) with pixels having the value intensity.

find_cosme

```
find_cosme cold_sigma hot_sigma
```

This command applies an automatic detection of cold and hot pixels following the threshold written in arguments.

find_cosme_cfa

```
find_cosme_cfa cold_sigma hot_sigma
```

Same command as FIND_COSME but for monochromatic CFA images.

find_hot

```
find_hot filename cold_sigma hot_sigma
```

The command provides a list file `filename` (format text) in the working directory which contains the coordinates of the pixels which have an intensity `hot_sigma` times higher and `cold_sigma` lower than standard deviation, identified in the currently loaded image. We generally use this command on a master-dark file. The COSME command can apply this list of bad pixels to a loaded image, see also SEQCOSME to apply it to a sequence.

findstar

```
findstar
```

Detects stars in the image, based on parameters controlled by setfindstar. They are visualized by drawing an orange ellipse around each. See also the command CLEARSTAR.

fix_xtrans

```
fix_xtrans
```

Fixes the Fujifilm X-Trans Auto Focus pixels. Indeed, because of the phase detection auto focus system, the photosites used for auto focus get a little less light than the surrounding photosites. The camera compensates for this and increases the values from these specific photosites giving a visible square in the middle of the dark/bias frames.

fixbanding

```
fixbanding amount sigma
```

Try to remove the canon banding. Argument `amount` define the amouont of correction. Sigma defines a protection level of the algorithm, higher sigma gives higher protection.

fmedian

```
fmedian ksize modulation
```

Performs a median filter of size ksize $\times$ksize (ksize MUST be odd) to the original image with a modulation parameter modulation. The output pixel is computed as : out $=\text{mod} \times m + (1-\text{mod})\times\text{in}$, where m is the median-filtered pixel value. A modulation's value of 1 will apply no modulation.

fmul

```
fmul scalar
```

Multiplies the loaded image by the scalar given in argument.

gauss

```
gauss sigma
```

Applies to the working image a Gaussian blur (https://en.wikipedia.org/wiki/Gaussian_blur) with parameter `sigma`.

See also UNSHARP, the same with a blending parameter.

get

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

Get a setting value, using its variable name, or list all with `-a` (name and value list) or `-A` (detailed list).

See also the SET command to update values.

getref

```
getref sequencename
```

Prints information about the reference image of the sequence given in argument. First image has index 0.

ght

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

Generalised hyperbolic stretch based on the work of the ghsastro.co.uk (https://ghsastro.co.uk/) team.

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

- `B` defines the intensity of the stretch near the focal point, between -5 and 15
- `LP` defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail
- `SP` defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense
- `HP` defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted B, LP and SP default to 0.0 ad HP defaults to 1.0. An optional argument, either `-human`, `-even` or `-independent`, can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument `-sat` specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

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

grey_flat

```
grey_flat
```

The function equalizes the mean intensity of RGB layers in a CFA images. This is the same process used on flats during preprocessing when the option *equalize CFA* is used.

help

```
help [command]
```

Lists the available commands or help for one command.

histo

```
histo layer
```

Calculates the histogram of the image layer in memory and produces file histo_[layer name].dat in the working directory.

layer = 0, 1 or 2 with 0=red, 1=green and 2=blue.

iadd

```
iadd filename
```

Adds the image in memory to the image designed in argument. Please check that the image is in the working directory.

idiv

```
idiv filename
```

Divides the image in memory by the image given in argument. Please check that the image is in the working directory. See also fdiv.

imul

```
imul filename
```

Multiplies the image in memory by the image given in argument. Please check that the image is in the working directory.

inspector

```
inspector
```

Splits the current image in a nine-panel mosaic showing the image corners and the center for a closer inspection.

invght

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

Invert a generalised hyperbolic stretch. It provides the inverse transformation of GHT, if provided with the same parameters, to undo a GHT command, possibly returning to a linear image. It can also work the same way as GHT but for images in negative.

invmodasinh

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

Invert a modified asinh stretch. It provides the inverse transformation of MODASINH, if provided with the same parameters, to undo a MODASINH command, possibly returning to a linear image. It can also work the same way as MODASINH but for images in negative.

invmtf

```
invmtf low mid high [channels]
```

Inverse of the MTF (midtones transfer function) command.

isub

```
isub filename
```

Substracts the image in memory by the image given in argument. Please check that the image is in the working directory.

jsonmetadata

```
jsonmetadata FITS_file [-stats_from_loaded] [-nostats] [-out=]
```

Dumps metadata and statistics of the currently loaded image in JSON form. The file name is required, even if the image is already loaded. Statistics can be disabled by providing the `-nostats` option. A file containing the JSON data is created with default file name out.json and can be changed with the `-out=` option.

light_curve

```
light_curve sequencename channel [-autoring] { -at=x,y | -wcs=ra,dec } { -refat=x,y | -refwcs=ra,dec } ...
light_curve sequencename channel [-autoring] -ninastars=file
```

Analyse several stars with aperture photometry in a sequence of images and produce a light curve for one, calibrated by the others. The first coordinates, in pixels if `-at=` is used or in degrees if `-wcs=` is used, are for the star whose light will be plotted, the others for the reference stars.

Alternatively, a list of target and reference stars can be passed in the format of the NINA exolpanet plugin star list, with the `-ninastars=` option. Siril will verify that all reference stars can be used before actually using them. A data file is created in the current directory named light_curve.dat, gnuplot plots the result to a PNG image if available.

The ring radii for aperture photometry can either be configured in the settings or set to a factor of the reference image's FWHM if `-autoring` is passed.

See also SEQPSF for operations on single star.

linear_match

```
linear_match reference low high
```

Computes a linear function between a reference image and a target image already loaded in memory. The function is then applied to the target image to match it to the reference one. The algorithm will ignore all reference pixels whose values are outside of the [low, high] range. The target image is not saved after the call, see the save command.

link

```
 link basename [-start=index] [-out=directory]
```

Create a new sequence from all FITS images found in the current working directory by linking them to a new name starting with the basename given in argument. If no symbolic link could be created, files are copied. The `-out=` option changes the output directory to the provided argument.

linstretch

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

Stretches the image linearly to a new black point BP. There is one mandatory argument: `-BP=` provides the new black point to stretch to. The argument `channels` may optionally be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels. Optionally the parameter `-sat` may be used to apply the linear stretch to the image saturation channel. This argument only works if all channels are selected.

livestack

```
livestack filename
```

Process the provided image for live stacking. Only possible after START_LS. The process involves calibrating the incoming file if configured in START_LS, demosaicing if it's an OSC image, registering and stacking. The temporary result will be in the file `live_stack_00001.fit` until a new option to change it is added.

Note that the live stacking commands put Siril in a state in which it's not able to process other commands. After START_LS, only LIVESTACK, STOP_LS and EXIT can be called until STOP_LS is called to return Siril in its normal, non-live-stacking, state.

load

```
load filename
load filename.ext
```

Loads the image filename; it first attempts to load filename, then filename.fit and finally filename.fits and after, all supported format, aborting if none of these are found. These scheme is applicable to every Siril command implying reading files. Fits headers MIPS-HI and MIPS-LO are read and their values given to the current viewing levels. Writing a known extension at the end of filename will load the image filename.ext: this is used when numerous files have the same name but not the same extension.

Extensions supported are :

- *.fit, *.fits, *.fts
- *.bmp / *.jpg, *.jpeg / *.png / *.tif, *.tiff
- *.ppm, *.pgm
- *.pic (IRIS file)

log

```
log
```

Computes and applies a logarithmic scale to the current image, using the following formula: log(1 - (value - min) / (max - min)), with min and max being the minimum and maximum pixel value for the channel.

ls

```
ls
```

Lists files and directories in the working directory.

makepsf

```
makepsf load filename
makepsf save
makepsf clear
makepsf blind [-l0] [-si] [-multiscale] [-lambda=] [-comp=] [-ks=]
makepsf stars [-sym] [-ks=]
makepsf manual { -gaussian | -moffat | -disc | -airy } [-fwhm=] [-angle=] [-ratio=] [-beta=] [-dia=] [-fl=] [-wl=] [-pixelsize=] [-obstruct=] [-ks=]
```

Generates a PSF for use with deconvolution, any of the three methods exposed by RL, SB or WIENER commands. One of the following must be given as the first argument:

- `load` loads a PSF from a file
- `save` saves the current PSF. This may be done in any format that Siril has been compiled with support for, but it must be square and should ideally be odd. No additional argument is required: the PSF file will be named based on the name of the open file or sequence

- `clear` clears the existing PSF, no additional argument required
- `blind` blind estimate of the PSF. The following optional arguments may be provided: `-l0` uses the l0 descent method, `-si` uses the spectral irregularity method, `-multiscale` configures the l0 method to do a multi-scale PSF estimate, `-lambda=` provides the regularization constant
- `stars` generates a PSF based on measured stars from the image. The only optional parameter is `-sym`, which configures the PSF to be symmetric
- `manual` generates a PSF manually based on a function and parameters. One of `-gaussian`, `-moffat`, `-disc` or `-airy` must be provided to specify the PSF function.
  - For Gaussian or Moffat PSF the optional arguments `-fwhm=`, `-angle=` and `-ratio=` may be provided.
  - For Moffat PSF the optional argument `-beta=` may also be provided. If these values are omitted, they default to the same values as in the deconvolution dialog.
  - For disc PSFs only the argument `-fwhm=` is required, which for this function is used to set the *diameter* of the PSF.
  - For Airy PSFs the following arguments may be provided: `-dia=` (sets the telescope diameter in mm), `-fl=` (sets the telescope focal length in mm), `-wl=` (sets the wavelength to calculate the Airy diffraction pattern for in nm), `-pixelsize=` (sets the sensor pixel size in μm), `-obstruct=` (sets the central obstruction as a percentage of the overall aperture area). If these parameters are not provided, wavelength will default to 525nm and central obstruction will default to 0%. Siril will attempt to read the others from the open image, but some imaging software may not provide all of them in which case you will get bad results, and note the metadata may not be populated for SER format videos.

For any of the above PSF generation options the optional argument `-ks=` may be provided to set the PSF dimension.

merge

```
merge seq1 seq2 [... seqn] newseq
```

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 colour image from 4 separate images containing the data from Bayer subchannels CFA0, CFA1, CFA2 and CFA3. (The corresponding command to split the CFA pattern into subchannels is `split_cfa`.) This function can be used as part of a workflow applying some processing to the individual Bayer subchannels prior to demosaicing. The fifth parameter `bayerpattern` specifies the Bayer matrix pattern to recreate: `bayerpattern` should be one of 'RGGB', 'BGGR', 'GRBG' or 'GBRG'.

mirrorx

```
mirrorx [-bottomup]
```

Rotates the image around a vertical axis. Option `-bottomup` will only flip it if it's not already bottom-up

mirrorx_single

```
mirrorx_single image
```

Flips the image about the vertical axis, only if needed (if it's not already bottom-up). It takes the image file name as argument, allowing it to avoid reading image data entirely if no flip is required.

mirrory

```
mirrory
```

Rotates the image around an horizontal axis.

modasinh

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

Modified arcsinh stretch based on the work of the ghsastro.co.uk (https://ghsastro.co.uk/) team. This is similar to the simple asinh stretch but offers additional parameters for fine tuning.

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

- `LP` defines a shadow preserving range between 0 and SP where the stretch will be linear, preserving shadow detail
- `SP` defines the symmetry point of the stretch, between 0 and 1, which is the point at which the stretch will be most intense
- `HP` defines a region between HP and 1 where the stretch is linear, preserving highlight details and preventing star bloat.

If omitted LP and SP default to 0.0 ad HP defaults to 1.0.\nAn optional argument, either `-human`, `-even` or `-independent`, can be passed to select either human-weighted or even-weighted luminance or independent colour channels for colour stretches. The argument is ignored for mono images. Alternatively, the argument `-sat` specifies that the stretch is performed on image saturation - the image must be color and all channels must be selected for this to work.

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

mtf

```
mtf low midtone high [channels]
```

Applies midtone transfer function to the current loaded image. Three parameters are needed, `low`, `midtones` and `high` where midtones balance parameter defines a nonlinear histogram stretch in the [0,1] range. Optionally, the parameter `channels` may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

neg

```
neg
```

Changes pixel values of the currently loaded image to a negative view, like 1-value for 32 bits, 65535-value for 16 bits. This does not change the display mode.

new

```
new width height nb_layers
```

Creates a new image filled with zeros with a size of `width x height`. The image is in 16-bit format, and it contains `nb_layers` layers, `nb_layers` being 1 or 3. It is not saved, but displayed and can be saved afterwards.

nomad

```
nomad [limit_magnitude] [-catalog=] [-photo]
```

If local star catalogues have been installed and currently loaded image has WCS information (= has been plate solved), display the stars with photometric information with at their expected positions, down to the provided magnitude, or 13 by default. An alternate catalog can be specified with **-catalog=**, taking values tycho2, nomad, gaia, ppmxl, brightstars, apass. By default stars with no B-V information will be kept, they can be excluded by specifying **-photo** (as required for the photometric color calibration (PCC)).

Use CLEARSTAR to clear the drawn stars. This is only available from the GUI of Siril.

nozero

```
nozero level
```

Replaces null values by level values. Useful before an idiv or fdiv operation.

offset

```
offset value
```

Adds the constant value to the current image. This constant can take a negative value. As Siril uses unsigned fit files, if the intensity of the pixel become negative its value is replaced by 0 and by 65535 (for a 16-bit file) if the pixel intensity overflows. To check the minimum and maximum intensities values, click on the Auto level button and note the low and high threshold.

parse

```
parse str [-r]
```

Parses the string `str` using the information contained in the header of the image currently loaded.

Option `-r` specifies the string is to be interpreted in read mode. In read mode, all wilcards defined in string `str` are used to find a file name matching the pattern. Otherwise, default mode is write mode and wildcards, if any, are removed from the string to be parsed.

If `str` starts with *lib* prefix, it will be recognized as a reserved keyword and look for the string stored in gui_prepro.dark_lib, gui_prepro.flat_lib or gui_prepro.use_bias_lib for dark, flat or bias respectively.

pcc

```
pcc [image_center_coords] [-noflip] [-platesolve] [-focal=] [-pixelsize=] [-limitmag=[+-]] [-catalog=] [-downscale]
```

Run the Photometric Color Correction on the loaded image.

If the image has already been plate solved, the PCC can reuse the astrometric solution, otherwise, or if WCS or other image metadata is erroneous or missing, arguments for the plate solving must be passed:

- the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space
- focal length and pixel size can be passed with `-focal=` (in mm) and `-pixelsize=` (in microns), overriding values from image and settings
- you can force the plate solving to be remade using the `-platesolve` flag.

Unless `-noflip` is specified, if the image is detected as being upside-down, it will be flipped if a plate solving is run. For faster star detection in big images, down-sampling the image is possible with `-downscale`.

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

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

platesolve

```
platesolve [image_center_coords] [-noflip] [-platesolve] [-focal=] [-pixelsize=] [-limitmag=[+-]] [-catalog=] [-
localasnet] [-downscale]
```

Plate solve the loaded image.

If the image has already been plate solved nothing will be done, unless the `-platesolve` argument is passed to force a new solve. If WCS or other image metadata is erroneous or missing, arguments must be passed:

- the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space (not mandatory for astrometry.net)
- focal length and pixel size can be passed with `-focal=` (in mm) and `-pixelsize=` (in microns), overriding values from image and settings.

Unless `-noflip` is specified and the image is detected as being upside-down, the image will be flipped. For faster star detection in big images, down-sampling the image is possible with `-downscale`.

Images can be either plate solved by Siril using a star catalogue and the global registration algorithm or by astrometry.net's local solve-field command (enabled with `-localasnet`).

The following options apply to Siril's plate solve only.

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

The choice of the star catalog is automatic unless the `-catalog=` option is passed: if local catalogs are installed, they are used, otherwise the choice is based on the field of view and limit magnitude. If the option is passed, it forces the use of the remote catalog given in argument, with possible values: `tycho2`, `nomad`, `gaia`, `ppmxl`, `brightstars`, `apass`.

pm

```
pm "expression" [-rescale [min max]]
```

This command evaluates the expression given in argument as in PixelMath tool. The full expression must be between double quotes and variables (that are image names located in the working directory in that case) must be surrounded by the token $, e.g. `"$image1$ * 0.5 + $image2$ * 0.5"`. A maximum of 10 images can be used in the expression. The result can be rescaled with the option `-rescale` followed by `low` and `high` values in the range [0, 1], defaulting to 0 and 1.

preprocess

Renamed CALIBRATE.

preprocess_single

Renamed CALIBRATE_SINGLE.

psf

```
psf [channel]
```

Performs a PSF (Point Spread Function) on the selected star and displays the results. If provided, the `channel` argument selects the image channel on which the star will be analyzed. It can be omitted for monochrome images or when run from the GUI with one of the channels active in the view.

The command will output:

- The centroid coordinates (x0 and y0) in pixel units, which is the position of the center of symmetry of the fitted PSF.
- The FWHM on the X and Y axis.
- The rotation angle of the X axis with respect to the centroid coordinates.
- The average local background.
- The maximal intensity of the star: this is the peak value of the fitted function, located at the centroid coordinates x0 and y0.
- The relative magnitude of the star.
- The RMSE. This is an estimate of fitting quality. The smaller the RMSE is, the better the function is fitted.

To be relevant, the selection MUST be done on a non-saturated star.

register

```
register sequence [-2pass] [-noout] [-drizzle] [-prefix=] [-minpairs=] [-transf=] [-layer=] [-maxstars=] [-nostarlist]
[-interp=] [-selected]
```

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

The `-2pass` and `-noout` options will only compute the transforms but not generate the transformed images. In that case, use the SEQAPPLYREG command to generate the transformed sequence used during stacking. In other cases, transformed sequence is created with a name prefixed with `r_` unless otherwise specified with the `-prefix=` option.

The reference frame, if not previously chosen with SETREF or from the GUI, will be set to the first image of the sequence. This can cause inaccurate registration if this first image is not of good quality (bad focus or tracking, cloud pass...). The option `-2pass` added in version 1.2 adds a preliminary pass to the registration, first detecting stars and choosing the best reference frame using a function based on FWHM and the number of stars, then computing the transforms of all frames relative to this new reference. The only drawback is that this makes the registration process a little slower because images must be read twice, once by this command and once by SEQAPPLYREG. `-2pass` never produces an output, so implies `-noout`. `-noout` alone uses the single pass, first image as reference, registration. `-nostarlist` disables saving the star lists to disk for caching purposes.

The option `-transf=` specifies the type of transformation to compute:

- `shift`, a 2 degrees of freedom (dof), X and Y translation, rigid transform (can be used with `-noout` to not create a new sequence, and still be stacked directly)
- `similarity`, a 4 dof transform, X and Y translation, a scale and rotation
- `affine`, a 6 dof transform, X and Y translation, two scales, one shear and rotation
- `homography` (default), a 8 dof warping transform.

The option `-drizzle` activates the sub-pixel stacking by up-scaling by 2 the generated images or by setting a flag that will proceed to the up-scaling during stacking if `-transf=shift -noout` are passed.

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

The pixel interpolation method used during image transformation can be specified with the `-interp=` keyword followed by one of the methods in the list `{ no[ne] | ne[arest] | cu[bic] | la[nczos4] | li[near] | ar[ea] }`. If `none` is passed, the transformation is forced to `shift` and a pixel-wise shift is applied to each image without any interpolation. This option is unused if the output sequence is not created.

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

reloadscripts

```
reloadscripts
```

Rescans the scripts folders and updates scripts menu.

requires

```
requires x.y.z
```

This function returns an error if the version of Siril is older than the one passed in argument.

resample

```
resample { factor | -width= | -height= } [-interp=] [-noclamp]
```

Resamples image, either with a factor `factor` or for the target width or height provided by either of `-width=` or `-height=`. This is generally used to resize images, a factor of 0.5 divides size by 2.

In the graphical user interface, we can see that several interpolation algorithms are proposed.

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

Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the `-noclamp` argument.

rgbcomp

```
rgbcomp red green blue [-out=result_filename]
rgbcomp -lum=image { rgb_image | red green blue } [-out=result_filename]
```

Create an RGB composition using three independent images, or an LRGB composition using the optional luminance image and three monochrome images or a color image. Result image is called composed_rgb.fit or composed_lrgb.fit unless another name is provided in the optional argument.

rgradient

```
rgradient xc yc dR dalpha
```

Creates two images, with a radial shift (`dR` in pixels) and a rotational shift (`dalpha` in degrees) with respect to the point (`xc`, `yc`). Between these two images, the shifts have the same amplitude, but an opposite sign. The two images are then added to create the final image. This process is also called Larson Sekanina filter.

rl

```
rl [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

Restores an image using the Richardson-Lucy deconvolution method.

The number of iterations is provide by `-iters` (the default is 10).

The type of regularization can be set with `-tv` for Total Variation, or `-fh` for the Frobenius norm of the Hessian matrix (the default is none) and `-alpha=` provides the regularization strength (lower value = more regularization, default = 3000).

By default the gradient descent method is used with a default step size of 0.0005, however the multiplicative method may be specified with `-mul`.

The stopping criterion may be activated by specifying a stopping limit with `-stop=`.

rmgreen

```
rmgreen [-nopreserve] [type] [amount]
```

Applies a chromatic noise reduction filter. It removes green tint in the current image. This filter is based on PixInsight's SCNR and it is also the same filter used by HLVG plugin in Photoshop.

Lightness is preserved by default but this can be disabled with the `-nopreserve` switch.

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

rotate

```
rotate degree
```

Rotates the image by an angle of `degree` value. The option `-nocrop` can be added to avoid cropping to the image size (black borders will be added).

Note: if a selection is active, i.e. by using a command BOXSELECT before ROTATE, the resulting image will be a rotated crop. In this particular case, the option `-nocrop` will be ignored if passed.

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

Clamping of the bicubic and lanczos4 interpolation methods is the default, to avoid artefacts, but can be disabled with the `-noclamp` argument.

rotatepi

```
rotatepi
```

Rotates the image of an angle of 180° around its center. This is equivalent to the command `rotate 180` or `rotate -180`.

satu

```
satu amount [background_factor [hue_range_index]]
```

Enhances the color saturation of the loaded image. Try iteratively to obtain best results.

- `amount` can be a positive number to increase color saturation, negative to decrease it, 0 would do nothing, 1 would increase it by 100%
- `background_factor` is a factor to (median + sigma) used to set a threshold for which only pixels above it would be modified. This allows background noise to not be color saturated, if chosen carefully. Defaults to 1. Setting 0 disables the threshold.
- `hue_range_index` can be [0, 6], meaning: 0 for pink to orange, 1 for orange to yellow, 2 for yellow to cyan, 3 for cyan, 4 for cyan to magenta, 5 for magenta to pink, 6 for all (default).

save

```
save filename
```

Saves current image to filename.fit. Fits headers MIPS-HI and MIPS-LO are added with values corresponding to the current viewing levels. `filename` can contain a path as long as the directory already exists.

savebmp

```
savebmp filename
```

Saves current image under the form of a bitmap file with 8bits per channel: filename.bmp (BMP 24 bits).

savejpg

```
savejpg filename [quality]
```

Saves current image into a JPG file. You have the possibility to adjust the quality of the compression. A value 100 for quality parameter offers best fidelity while a low value increases the compression ratio. If no value is specified, it holds a value of 100. This command is very usefull to share an image in the jpeg format on the forums for example.

savepng

```
savepng filename
```

Saves current image as a PNG file, with 16 bits per channel if the loaded image is 16 or 32 bits, and 8 bits per channel if the loaded image is 8 bits.

savepnm

```
savepnm filename
```

Saves current image under the form of a Netpbm file format with 16 bits per channel. The extension of the output will be filename.ppm for RGB image and filename.pgm for gray-level image. More details about the Netpbm format here (https://en.wikipedia.org/wiki/Netpbm_format).

savetif

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

Saves current image as an uncompressed TIFF file with 16bits per channel. The `.tif` suffix is added to the `filename`. The option `-astro` allows saving in Astro-tiff format. The option `-deflate` enables TIFF compression.

savetif32

```
savetif32 filename [-astro] [-deflate]
```

Same command as SAVETIF but the output file is saved in 32bits per channel. The option `-astro` allows saving in Astro-tiff format. The option `-deflate` enables TIFF compression.

savetif8

```
savetif8 filename [-astro] [-deflate]
```

Same command as SAVETIF but the output file is saved in 8bits per channel. The option `-astro` allows saving in Astro-tiff format. The option `-deflate` enables TIFF compression.

sb

```
sb [-alpha=] [-iters=]
```

Restores an image using the Split Bregman deconvolution method. The number of iterations is provide by `-iters` (the default is 1). The regularization factor `-alpha=` provides the regularization strength (lower value = more regularization, default = 3000).

select

```
select sequencename from to
```

This command allows easy mass selection of images in the sequence `sequencename` (from - to, to included). Examples:

```
select . 0 0
```

selects the first of the currently loaded sequence

```
select sequencename 1000 1200
```

selects 201 images starting from number 1000 in sequence named sequencename

The second number can be greater than the number of images to just go up to the end.

See UNSELECT.

seqapplyreg

```
seqapplyreg sequencename [-drizzle] [-interp=] [-layer=] [-framing=] [-prefix=] [filtering options]
```

Applies geometric transforms on images of the sequence given in argument so that they may be superimposed on the reference image, using registration data previously computed and optionally excluding images from the sequence using the *filtering options* documented in the STACK command reference.

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

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

The interpolation method can be specified with the `-interp=` keyword followed by one of the methods in the list `{ no[ne] | ne[arest] | cu[bic] | la[nczos4] | li[near] | ar[ea] }`. If `none` is passed, the transformation is forced to shift and a pixel-wise shift is applied to each image without any interpolation.

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

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

seqclean

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

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

You can specify to clear only registration, statistics and/or selection with `-reg`, `-stat` and `-sel` options respectively. All are cleared if no option is passed.

seqcosme

```
seqcosme sequencename [filename].lst [-prefix=]
```

Same command as COSME but for the the sequence `sequencename`.

The output sequence name starts with the prefix "cosme_" unless otherwise specified with option `-prefix=`. Only selected images in the sequence are processed.

seqcosme_cfa

```
seqcosme_cfa sequencename [filename].lst [-prefix=]
```

Same command as COSME_CFA but for the the sequence `sequencename`.

The output sequence name starts with the prefix "cosme_" unless otherwise specified with option `-prefix=`. Only selected images in the sequence are processed.

seqcrop

```
seqcrop sequencename x y width height [-prefix=]
```

Crops the sequence given in the `sequencename` argument. The output sequence name starts with the prefix "cropped_" unless otherwise specified with `-prefix=` option. Only selected images in the sequence are processed.

seqextract_Green

```
seqextract_Green sequencename [-prefix=]
```

Same command as EXTRACT_GREEN but for the sequence `sequencename`. The output sequence name starts with the prefix "Green_" unless otherwise specified with option `-prefix=`. Only selected images in the sequence are processed.

seqextract_Ha

```
seqextract_Ha sequencename [-prefix=]
```

Same command as EXTRACT_HA but for the sequence `sequencename`. The output sequence name starts with the prefix "Ha_" unless otherwise specified with option `-prefix=`. Only selected images in the sequence are processed.

seqextract_HaOIII

```
seqextract_HaOIII sequencename
```

Same command as EXTRACT_HAOIII but for the sequence `sequencename`. The output sequences names start with the prefixes "Ha_" and "OIII_". Only selected images in the sequence are processed.

seqfind_cosme

```
seqfind_cosme sequencename cold_sigma hot_sigma [-prefix=]
```

Same command as FIND_COSME but for the given sequence. The output sequence name starts with the prefix "cc_" unless otherwise specified with `-prefix=` option. Only selected images in the sequence are processed.

seqfind_cosme_cfa

```
seqfind_cosme_cfa sequencename cold_sigma hot_sigma [-prefix=]
```

Same command as FIND_COSME_CFA but for the given sequence. The output sequence name starts with the prefix "cc_" unless otherwise specified with `-prefix=` option. Only selected images in the sequence are processed.

seqfindstar

```
seqfindstar sequencename [-layer=] [-maxstars=]
```

Same command as FINDSTAR but for the sequence `sequencename`. The option `-out=` is not available for this process as all the star list files are saved with the default name `sequencename_seqnb.lst`.

seqfixbanding

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

Same command as FIXBANDING but for the sequence `sequencename`. The output sequence name starts with the prefix "unband_" unless otherwise specified with `-prefix=` option. `-vertical` option enables to perform vertical banding removal.

seqght

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

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

seqheader

```
seqheader sequencename keyword
```

Prints the FITS header value for the given key for all images in the sequence.

seqinvght

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

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

seqinvmodasinh

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

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

seqlinstretch

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

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

seqmerge_cfa

```
seqmerge_cfa sequencename bayerpattern [-prefixin=] [-prefixout=]
```

Same command as MERGE_CFA but for the sequence `sequencename`. The Bayer pattern to be reconstructed must be provided as the second argment as one of RGGB, BGGR, GBRG or GRBG. The input filenames conain the identifying prefix "CFA_" and a number unless otherwise specified with `-prefix=` option. Note: all 4 sets of input files `must` be present and `must` be consistently named, the only difference being the number after the identifying prefix. The output sequence name starts with the prefix "mCFA_" and a number unless otherwise specified with `-prefix=` option.

seqmodasinh

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

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

seqmtf

```
seqmtf sequencename low midtone high [channels] [-prefix=]
```

Same command as MTF but for the sequence sequencename. The output sequence name starts with the prefix "mtf_" unless otherwise specified with `-prefix=` option. Only selected images in the sequence are processed. Optionally, the parameter `channels` may be used to specify the channels to apply the stretch to: this may be R, G, B, RG, RB or GB. The default is all channels.

seqplatesolve

```
seqplatesolve sequencename [image_center_coords] [-noflip] [-platesolve] [-focal=] [-pixelsize=] [-limitmag=[+-]] [-
catalog=] [-downscale]
```

Plate solve a sequence. A new sequence will be created with the prefix "ps_".

If images have already been plate solved they will just be copied, unless the `-platesolve` argument is passed to force a new solve. If WCS or other image metadata are erroneous or missing, arguments must be passed:

- the approximate image center coordinates can be provided in decimal degrees or degree/hour minute second values (J2000 with colon separators), with right ascension and declination values separated by a comma or a space (not mandatory for astrometry.net)
- focal length and pixel size can be passed with `-focal=` (in mm) and `-pixelsize=` (in microns), overriding values from images and settings.

Unless `-noflip` is specified, if images are detected as being upside-down, they will be flipped. For faster star detection in big images, down-sampling the image is possible with `-downscale`.

Images can be either plate solved by Siril using a star catalogue and the global registration algorithm or by astrometry.net's local solve-field command (enabled with `-localasnet`).

The following options apply to Siril's plate solve only.

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

The choice of the star catalog is automatic unless the `-catalog=` option is passed: if local catalogs are installed, they are used, otherwise the choice is based on the field of view and limit magnitude. If the option is passed, it forces the use of the remote catalog given in argument, with possible values: `tycho2`, `nomad`, `gaia`, `ppmxl`, `brightstars`, `apass`.

seqpsf

```
seqpsf [sequencename channel { -at=x,y | -wcs=ra,dec }]
```

Same command as PSF but works for sequences. This is similar to the one-star registration, except results can be used for photometry analysis rather than aligning images.

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. By default, it will run with parallelisation activated; if registration data already exists for the sequence, they will 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.

The *follow star* option can then be activated by going in the registration tab, selecting the one-star registration and checking the *follow star movement* box (default in headless if no registration data is available).

Results will be displayed in the Plot tab, from which they can also be exported to a comma-separated values (CSV) file for external analysis.

When creating a light curve, the first star for which seqpsf has been run, marked 'V' in the display, will be considered as the variable star. All others are averaged to create a reference light curve subtracted to the light curve of the variable star.

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 in headless, with `-at=` allowing coordinates in pixels to be provided for the target star and `-wcs=` allowing J2000 equatorial coordinates to be provided.

seqrl

```
seqrl sequencename [-alpha=] [-iters=] [-stop=] [-gdstep=] [-tv] [-fh] [-mul]
```

The same as the RL deconvolution command, but applies to a sequence which must be specified as the first argument.

seqsb

```
seqsb sequencename [-alpha=] [-iters=]
```

The same as the SB deconvolution command, but applies to a sequence which must be specified as the first argument.

seqsetmag

```
seqsetmag magnitude
```

This command is only valid after having run seqpsf or its graphical counterpart (select the area around a star and launch the psf analysis for the sequence, it will appear in the graphs). This command has the same goal as setmag but recomputes the reference magnitude for each image of the sequence where the reference star has been found. When running the command, the last star that has been analysed will be considered as the reference star. Displaying the magnitude plot before typing the command makes it easy to understand. To reset the reference star and magnitude offset, see sequnsetmag.

seqsplit_cfa

```
seqsplit_cfa sequencename [-prefix=]
```

Same command as SPLIT_CFA but for the sequence `sequencename`. The output sequence name starts with the prefix "CFA_" and a number unless otherwise specified with `-prefix=` option. Only selected images in the sequence are processed.

Limitation: the sequence always outputs a sequence of FITS files, no matter the type of input sequence.

seqstarnet

```
seqstarnet sequencename [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

This command calls Starnet to remove stars from the current sequence. The first argument must be the sequence from which to remove stars: all the other available arguments are the same as for the STARNET command.

seqstat

```
seqstat sequencename output [option] [-cfa]
```

Same command as STAT bit for sequence `sequencename`. Only selected images in the sequence are processed. The output is saved in a csv file given in second argument, the meaning of each value is explained in the statistics documentation page (https://free-astro.org/index.php?title=Siril:Statistics).

The optional parameter can be `basic`, `main` or `full`. Default is `main`.

- `basic` includes mean, median, sigma, bgnoise, min and max
- `main` includes `basic` with the addition of avgDev, MAD and the square root of BWMV
- `full` includes `main` with the addition of location and scale.

If `-cfa` is passed and the images are CFA, statistics are made on per-filter extractions.

seqsubsky

```
seqsubsky sequencename { -rbf | degree } [-nodither] [-samples=20] [-tolerance=1.0] [-smooth=0.5] [-prefix=]
```

Same command as SUBSKY but for the sequence `sequencename`. Dithering, required for low dynamic gradients, can be disabled with `-nodither`. The output sequence name starts with the prefix "bkg_" unless otherwise specified with `-prefix=` option. Only selected images in the sequence are processed.

sequnsetmag

```
sequnsetmag
```

Resets the magnitude calibration and reference star for the sequence. See SETMAGSEQ.

seqtilt

```
seqtilt [sequencename]
```

Same command as TILT but for the loaded sequence (from the GUI) or the sequence `sequencename`. It generally gives better result.

seqwiener

```
seqwiener sequencename [-alpha=]
```

The same as the WIENER deconvolution command, but applies to a sequence which must be specified as the first argument.

set

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

Update a setting value, using its variable name, with the given value, or a set of values using an existing ini file with `-import=` option.

See GET to get values or the list of variables.

set16bits

```
set16bits
```

Forbid images to be saved with 32 bits per channel on processing, use 16 instead.

set32bits

```
set32bits
```

Allow images to be saved with 32 bits per channel on processing.

setcompress

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

Defines if images are compressed or not: 0 means no compression. If compression is enabled, the type must be explicitly written in the option `-type=` (`rice`, `gzip1`, `gzip2` and `hcompress`). Associated to the compression, the quantization value must follow [0, 256] and in the case of `hcompress`, the hcompress scale factor [0, 256] as well. For example, `setcompress 1 -type=rice 16` set the rice compression with a quantization of 16.

setcpu

```
setcpu number
```

Defines the number of processing threads used for calculation. Can be as high as the number of virtual threads existing on the system, which is the number of CPU cores or twice this number if hyper-threading (Intel HT (https://en.wikipedia.org/wiki/Hyper-threading)) is available. The default value is the maximum number of threads available, so this should mostly be used to limit processing power. See also SETMEM.

Warning: this command does not persist over siril restarts, contrary to SETMEM which is saved in settings.

setext

```
 setext extension
```

Sets the extension used and recognized by sequences. The argument `extension` can be `fit`, `fts` or `fits`.

setfindstar

```
setfindstar [reset] [-radius=] [-sigma=] [-roundness=] [-focal=] [-pixelsize=] [-convergence=] [ [-gaussian] | [-
moffat] ] [-minbeta=] [-relax=on|off] [-minA=] [-maxA=] [-maxR=]
```

Defines stars detection parameters for FINDSTAR, REGISTER, PCC, PLATESOLVE commands and the like.

Passing no parameter lists the current values. Passing `reset` resets all values to defaults. You can then still pass values after this keyword.

The threshold for star detection is computed as the median of the image (which represents in general the background level) plus k times sigma, sigma being the standard deviation of the image (which is a good indication of the noise amplitude). If you have many stars in your images and a good signal/noise ratio, it may be a good idea to increase this value to speed-up the detection and false positives.

It is recommended to **test the values** used for a sequence with Siril's GUI, available in the dynamic PSF toolbox from the analysis menu. It may improve registration quality to increase the parameters, but it is also important to be able to detect several tens of stars in each image.

`-radius=` defines the radius of the initial search box and must be between 3 and 50.
`-sigma=` defines the threshold above noise and must be greater or equal to 0.05.
`-roundness=` defines minimum star roundness and must between 0 and 0.95. A value of 1 would keep only

perfectly round stars, a value of 0.5, the default, would keep stars twice as wide as high. `-maxR` allows an upper bound to roundness to be set, to visualize only the areas where stars are significantly elongated, do not change for registration.
`-minA` and `-maxA` define limits for the minimum and maximum amplitude of stars to keep, normalized between 0 and 1 (0 for both disables this check).
`-focal=` defines the focal length of the telescope.
`-pixelsize=` defines the pixel size of the sensor.
`-gaussian` and `-moffat` configure the solver model to be used (Gaussian is the default). If Moffat is selected, `-minbeta=` defines the minimum value of beta for which candidate stars will be accepted and must be greater or equal to 0.0 and less than 10.0.
`-convergence=` defines the number of iterations performed to fit PSF and should be set between 1 and 3 (more tolerant).
`-relax=` relaxes checks that are done on star candidates to assess if they are stars or not, to allow objects not shaped like stars to still be accepted (off by default).

setmag

```
setmag magnitude
```

Calibrates the magnitude by selecting a star and giving the known apparent magnitude. All PSF computations will return the calibrated apparent magnitude afterwards, instead of an apparent magnitude relative to ADU values. To reset the magnitude constant see UNSETMAG.

setmem

```
setmem ratio
```

Sets a new ratio of free memory on memory used for stacking. Value should be between 0 and 1, depending on other activities of the machine. A higher ratio should allow siril to stack faster. Introduced in 0.9.11. See also SETCPU.

setphot

```
setphot [-inner=20] [-outer=30] [-aperture=10] [-force_radius=no] [-gain=2.3] [-min_val=0] [-max_val=60000]
```

Gets or sets photometry settings, mostly used by seqpsf. If arguments are provided, they will update the settings. None are mandatory, any can be provided, default values are shown in the command's syntax. At the end of the command's execution, the active configuration will be printed. Aperture is dynamic unless forced, the `aperture` value from settings is not used if dynamic, FWHM is used instead. Gain is used only if not available from the FITS header. See the documentation on these parameters in the aperture photometry page.

setref

```
setref sequencename image
```

Sets the reference image of the sequence given in first argument.

show

```
show [-clear] [{ -list=file.csv | [name] RA Dec }]
```

Shows a point on the loaded plate solved image using the temporary user annotations catalogue, based on its equatorial coordinates. The `-reset` option clears this catalogue first and can be used alone. Several points can be passed using a CSV file with the option `-file=` containing a name, ra and dec. This is only available from the GUI of Siril.

solsys

```
solsys [-mag=20.0]
```

Search and display solar system objects in the current loaded and plate solved image. Use `-mag=` to change the limit magnitude which defaults to 20. This research has made use of IMCCE's SkyBoT VO tool (https://ui.adsabs.harvard.edu/abs/2006ASPC..351..367B/abstract)

split

```
split file1 file2 file3 [-hsl | -hsv | -lab]
```

Splits the currently loaded color image into three distincts files (one for each color) and save them in file1 file2 and file3 FITS files. A last argument can optionally be supplied, `-hsl`, `-hsv` or `lab` to perform an HSL, HSV or CieLAB extraction. If no option are provided, the extraction is of RGB type.

split_cfa

```
split_cfa
```

Splits the currently loaded CFA image into four distinct images (one for each filter of the Bayer matrix) and save them in FITS files. See SEQSPLIT_CFA for the same operation on a sequence.

Not supported for XTRANS sensors.

stack

```
 stack seqfilename
 stack seqfilename { sum | min | max } [filtering] [-output_norm] [-out=filename]
 stack seqfilename { med | median } [-nonorm, -norm=] [filtering] [-fastnorm] [-rgb_equal] [-output_norm] [-out=filename]
 stack seqfilename { rej | mean } [rejection type] [sigma_low sigma_high]  [-rejmap[s]] [-nonorm, -norm=] [filtering] [-fastnorm] [ -weight_from_noise | -weight_from_nbstack | -weight_from_wfwhm | -weight_from_nbstars ] [-rgb_equal] [-output_norm] [-out=filename]
```

with `filtering` being any of these options, in no particular order or number:

```
[-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]]
```

Stacks the `seqfilename` sequence, using options.

The allowed types are: `sum`, `max`, `min`, `med` or `median` (these two are the same), `mean` or `rej` (same here).

The rejection type is one of `n[one]`, `p[ercentile]`, `s[igma]`, `m[edian]`, `w[insorized]`, `l[inear]`, `g[eneralized]` or `[m]a[d]` for no rejection or Percentile, Sigma, Median, Winsorized, Linear-Fit, Generalized Extreme Studentized Deviate Test and k-MAD clipping. If omitted, the default (Winsorized) is used. The `sigma_low` and `high` parameters of rejection are mandatory if rejection is not set to `none`. Optionally, rejection maps can be created, showing where pixels were rejected in one (`-rejmap`) or two (`-rejmaps`, for low and high rejections) newly created images. Rejection map files will be named like the original file, with suffixes `_low+high_rejmap`, `_low_rejmap` or `_high_rejmap` depending on the case.

See the tooltips in the stacking tab for more information about the stacking methods and rejection types, or see the documentation.

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

- a numeric value for the worse image to keep depending on the type of data used (between 0 and 1 for roundness and quality, absolute values otherwise),
- a percentage of best images to keep if the number is followed by a `%` sign,
- or a k value for the k.sigma of the worse image to keep if the number is followed by a `k` sign.

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

If several filters are added to the command, only images that pass all the filters will be stacked. There is consequently no order. If a filter is badly declared, because it has no registration data or a too low threshold, nothing will be stacked.

Normalization can be enabled for median and mean stacking methods using the `-norm=`*`normalization`* option. The allowed normalization are: `add`, `addscale`, `mul` or `mulscale`. For other methods, or with the use of the `-nonorm` flag, normalization is disabled. The additional flag `-fastnorm` will enable the use of faster estimators for location (median) and scale (MAD) than the default IKSS, which in a few cases may give less good results. `-rgb_equal` will change how color images are normalized to equalize their own channels, making their background more gray than the usual green. This is useful if no pcc can be made, or if no unlinked autostretch is to be used.

Weighting the images in the mean computation (after rejection if enabled) is possible using either the noise level computed from each image (option `-weight_from_noise`), the FWHM weighted by the number of stars common with the reference image (option `-weight_from_wfwhm`), the number of stars in the image (option `-weight_from_nbstars`) (the last two computed during registration), or using the number of images previously used to stack the input images, indicated by the `STACKCNT` or `NCOMBINE` FITS header keywords (option `-weight_from_nbstack`), the latter being used for live or iterative stacking. These options can only be enabled with mean or rejection stacking, when normalization has been enabled.

`-output_norm` applies a normalization at the end of the stacking to rescale result in the [0, 1] range.

Stacked image for the sequence is created with the name provided in the optional argument `-out`, or with the name of the sequence suffixed "_stacked" and the configured FITS file extension. If a file with this name already exists, it will be overwritten without warning.

Note that this command was added in the 0.9.9 release, the filtering and output naming options were added in the 0.9.11 release and the output_norm in the 1.0.0, weighting options, rgb equalization and disabling the rejection were added in 1.2.0.

stackall

```
 stackall
 stackall { sum | min | max } [filtering]
 stackall { med | median } [-nonorm, norm=] [-filter-incl[uded]]
 stackall { rej | mean } [rejection type] [sigma_low sigma_high] [-nonorm, norm=] [filtering] [ -weight_from_noise | -
weight_from_wfwhm | -weight_from_nbstars | -weight_from_nbstack ] [-rgb_equal] [-out=filename]
```

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

```
[-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]]
```

Opens all sequences in the current working directory (CWD) and stacks them with the optionally specified stacking method. See STACK command for options description.

Stacked images for each sequence are created with the suffix "_stacked" and the configured FITS file extension.

Note that most options for this command were introduced in the 0.9.8 release, the filtering options were introduced in the 0.9.11 release and the `output_norm` option in the 1.0.0 release.

starnet

```
starnet [-stretch] [-upscale] [-stride=value] [-nostarmask]
```

This command calls Starnet++ (https://www.starnetastro.com/) to remove stars from the current image.

Prerequisite: Starnet++ is an external program, with no affiliation with Siril, and must be installed correctly prior to first use of this command, with path to its installation directory correctly set in Preferences / Miscellaneous.

The starless image is loaded on completion, and a star mask image is created in the working directory unless the optional parameter `-nostarmask` is provided.

Optionally, parameters may be passed to the command:

- The option `-stretch` is for use with linear images and will apply a pre-stretch before running Starnet and the inverse stretch to the generated starless and starmask images.
- To improve star removal on images with very tight stars, the parameter `-upscale` may be provided. This will upsample the image by a factor of 2 prior to Starnet++ processing and rescale it to the original size afterwards, at the expense of more processing time.
- The optional parameter `-stride=value` may be provided, however the author of Starnet *strongly* recommends that the default stride of 256 be used.

More tips and tricks are available there (https://www.starnetastro.com/tips-tricks/).

start_ls

```
start_ls [-dark=filename] [-flat=filename]
```

Initialize a livestacking session, using the optional calibration files and wait for input files to be provided by the LIVESTACK command until STOP_LS is called.

Note that the live stacking commands put Siril in a state in which it's not able to process other commands. After START_LS, only LIVESTACK, STOP_LS and EXIT can be called until STOP_LS is called to return Siril in its normal, non-live-stacking, state.

stat

```
stat [-cfa] [main]
```

Returns global statistics of the current image, the basic list by default or the main list if `main` is passed. The meaning of each value is explained on the statistics documentation page (https://free-astro.org/index.php?title=Siril:Statistics). If a selection is made, statistics are computed within the selection. To compute the statistics on a sequence, see SEQSTAT. If `-cfa` is passed and the image is CFA, statistics are made on per-filter extractions.

stop_ls

```
stop_ls
```

Stop the live stacking session. Only possible after START_LS.

Note that the live stacking commands put Siril in a state in which it's not able to process other commands. After START_LS, only LIVESTACK, STOP_LS and EXIT can be called until STOP_LS is called to return Siril in its normal, non-live-stacking, state.

subsky

```
subsky { -rbf | degree } [-dither] [-samples=20] [-tolerance=1.0] [-smooth=0.5]
```

Computes a synthetic background gradient using either the polynomial function model of `degree` degrees or the RBF model (if `-rbf` is provided instead) and subtracts it from the image.

The number of samples per horizontal line and the tolerance to exclude brighter areas can be adjusted with the optional arguments. Tolerance is in mad units: median + tolerance * mad. Dithering, required for low dynamic gradients, can be enabled with `-dither`. For RBF, the additional smoothing parameter is also available.

synthstar

```
synthstar
```

Synthstar fixes bad stars. No matter how much coma, tracking drift or other distortion your stars have, if Siril's star finder routine can detect it, synthstar will fix it. To use intensive care, you may wish to manually detect all the stars you wish to fix. This can be done using the findstar console command or the Dynamic PSF dialog. If you have not run star detection, it will be run automatically with default settings. For best results synthstar should be run before stretching. The output of synthstar is a fully corrected synthetic star mask comprising perfectly round star PSFs (Moffat or Gaussian profiles depending on star saturation) computed to match the intensity, FWHM, hue and saturation measured for each star detected in the input image. This can then be recombined with the starless image to produce an image with perfect stars. No parameters are required for this command.

unclipstars

```
unclipstars
```

Re-profiles clipped stars to desaturate them, scaling the output so that all pixel values are <= 1.0.

threshlo, threshhi, thresh

These are threshold functions:

- `threshlo 40` replaces values below 40 with 40;
- `threshhi 1000` replaces values above 1000 with 1000;
- `thresh 40 1000` does both.

tilt

```
tilt
```

Computes the Sensor tilt as the fwhm difference between the best and worst corner truncated mean values. The `clear` option allows to clear the drawing.

unselect

```
unselect sequencename from to
```

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

unsetmag

```
unsetmag
```

Reset the magnitude calibration to 0. See SETMAG.

unsharp

```
unsharp sigma amount
```

Applies to the working image an unsharp mask, actually a Gaussian filtered (https://en.wikipedia.org/wiki/Gaussian_blur) image with parameter `sigma` and a blend with the parameter `amount` used as such: `out = in * (1 + amount) + filtered * (-amount)`.

See also GAUSS, the same without the blending.

visu

```
visu low high
```

Displays an image with low and high as the low and high threshold.

wavelet

```
wavelet plan_number type
```

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

wiener

```
wiener [-alpha=]
```

Restores an image using the Wiener deconvolution method. The parameter `-alpha=` provides the Gaussian noise modelled regularization factor.

wrecons

```
wrecons c1 c2 ... cn
```

Reconstructs to current image from the planes previously computed with wavelets and weighted with coefficients `c1, c2, ..., cn` according to the number of planes used for wavelet transform.

Citováno z „https://free-astro.org/index.php?title=Siril:Commands&oldid=8421“

Stránka byla naposledy editována 7. 4. 2023 v 15:13.