StarTools

Version 1.3.5

Unofficial User Manual

Edited by Stephen Statts

“There are as many schools of astroimage processing as there are astrophotographers.“

- Rogelio Bernal Andreo, Astrophotographer, DeepSkyColors.com

This manual is a modified version of the obsolete StarTools 1.0 manual

Last updated 05 May 2014

Please contact us through our website at http://www.startools.org if you find any mistakes, factual errors or copyright/usage violations in this manual.

Table of Contents

Introduction 8

Getting Started and system requirements 9

Processing with StarTools 11

An address to beginners and advanced users 11
Making your data StarTools friendly 12
What StarTools does and does not do 12

Interface 14

Track 17

Tracking Details 17

Modules 20

Open 23

LRGB 26

Develop 30

AutoDev 35

Contrast 38

HDR 41

Sharp 47

Deconvolution 52

Wipe 55

Bin 63

When should you consider software binning 63

Synth 67

Prerequisites 67
Star Mask 68
Telescope modeling 68

Star Synthesis 72
Examining an output sample 77
The ethics of using synthesised components in your image 79
Lens 80
Heal 83
Using Heal to process stars independently 84
Magic 91
How to make big/fat/white stars appear tighter 96
Filter 98
Example: Fringe and Halo Killer 99
Layer 105
Example 1-Integrating frames of different exposure lengths 110
Example 2-Rounding elongated stars 112
DENOISE 122
Denoise Preview Only Mode 125
BAND 126
Repair 127
How to make “eggy” stars appear rounder 128
LIFE 129
FLUX 131
Color 133
The Color module demystified 139
Crop 147
Mirror H 149
Mirror V 149
ROTATE 149
Masks 150

The Mask Editor.. 151
The Auto Feature.. 156
Example 1-Star Detection.. 158
Example 2: Noise Reduction.. 161
Example 3: Hot Pixel Removal.. 168
Acknowledgements.. 172
Appendix A.. 173
A Suggested StarTools Workflow.. 173

Introduction

Thank you for your interest in StarTools!

StarTools is an astronomical image post-processing program for Windows, MacOSX and Linux. StarTools sets itself apart from other processing software by focusing on the unique needs of amateur 'sidewalk astronomer' in urban areas.

StarTools tries to make up for a lack of professional equipment and imaging in less-thanideal urban skies. It does this by leveraging the abundance of cheap 64-bit CPU power and memory, as well as powerful off-the-shelf imaging hardware, such as Compact Digital Cameras, Digital Single-Lens Reflex cameras and webcams. Consequently StarTools contains features and algorithms that are not commonly found in software packages solely meant for professionals, with the latter type of astronomers typically having access to semi-professional equipment and dark sky sites.

StarTools can be used to post process a stacked image from start-to-finish, or may be used as part of a greater toolchain of image processing programs. Besides various unique and novel algorithms, StarTools also aims to offer equivalents of most of the various PhotoShop actions available on the web – all without the need to separately purchase a PhotoShop plug-in compatible host program.

The StarTools project was created for the love of astronomy and the desire to greatly lower the barrier for urban beginners interested in imaging the skies. A multi-platform approach, combined with not-for-profit pricing, promotes the author's ideal of 'astrophotography-for-all'. We hope you enjoy the software and that it will be instrumental in getting the most from the equipment you have.

Clear skies,

Ivo Jager, StarTools author

Getting Started and system requirements

StarTools, as with every other image processing suite, is memory and CPU intensive. A fast dual core processor with SSE3 is recommended as a minimum. A 64-bit Operating System is highly recommended, although 32-bit binaries are also available. 3Gb of memory or more is highly recommended for 32-bit Operating Systems, while 6Gb or more is recommended for 64-bit Operating Systems. Please note that the MacOSX version of StarTools will not run on PowerPC processors.

StarTools will run on Netbooks with Intel Atom processors, however limited CPU power and memory constraints will severely limit the size of the images that can be effectively processed.

The 32-bit version of StarTools will use a standard quality 32-bit signal path, whereas any 64-bit version of StarTools uses a high performance, high quality 64-bit engine. Therefore, if your hardware and operating system allows, please choose to run a 64-bit executable for your platform, as it contains some very important improvements over the 32-bit version. The 64-bit executable also allows StarTools to use more than 3 GB of memory, should your system have more available.

The disk space that is required for StarTools is very minimal; StarTools requires less than 2 Megabytes of disk space, which is less than a typical MP3 song. Despite its small size, StarTools is fully self-contained and does not depend on any external libraries or frameworks[1] and no further components need to be installed.

To install StarTools, simply download the ZIP archive and extract it to a location of your choice. If you purchased a license key, simply put the license key file in the same location as the StarTools executable and 'resources' file. If you do not have a license key, the executable will revert to being a demo without save capabilities. If you do have a license key but the program still reports itself as a demo version after you copied the key as outlined above, please contact us through our website at http://www.startools.org.

Please note that throughout this manual screenshots of the Linux version are shown, however, please be assured that StarTools on other platforms will appear virtually identical.

[1] StarTools for Linux should run on most distributions with X11 and GTK installed

Lastly, calibrating your screen is of the utmost importance. Failure to do so may invalidateyour processing efforts - incorrectly calibrated screens may cause you to perform incorrect colour correction, miss colour casts, gradients, noise and other artefacts. Subsequent viewing of your image on a correctly calibrated screen or in print may make these defects painfully visible. It is highly recommended to use the best screen to which you have access. At the very least your screen should be connected digitally (i.e. by DVI or HDMI cable). A dual monitor setup is ideal as it allows you to evaluate your image on different screens. Please note that most cheaply integrated laptop screens are usually not ideal as a primary screen for processing and a secondary screen is highly recommended.

Processing with StarTools

An address to beginners and advanced users

If you are a beginner, learn, become a member of the many forums on the Internet, read up on imaging and processing best practices and, most of all, have fun! StarTools will go a long way in satisfying your processing needs, especially if your equipment is relatively humble. StarTools is also an excellent companion for those using Global Rent-A-Scope[2] or LightBuckets[3] remote telescope rental services.

If you are a more advanced user, StarTools contains modules that rival (or best) some of the house hold names in astrophotography packages and plug-ins. In addition, StarTools sports some novel modules and approaches not found anywhere else, potentially giving you an edge.

Whether you're a beginner or an advanced user, don't be afraid to ask for help on the StarTools forums at http://www.startools.org/forum. For its modest size and goals, StarTools is a complex beast. Like any other complex piece of software, there will be a learning curve involved, though we've sought to minimise this learning curve wherever we had the chance – StarTools automates or abstracts a lot of tasks that may be quite tedious, time consuming or prone to error in other processing suites.

Lastly, regardless whether you're a beginner or advanced user, don't be afraid to experiment. More so, try to learn why algorithms and techniques work the way they do and how they can be of benefit to your work. It will make you that much more creative and agile when processing. Understanding the strengths and weaknesses of your image train from photon to pixel will make it easier to choose the right algorithms and parameters to get the most out of your precious data.

As alluded to by the quote from accomplished astrophotographer Rogelio Bernal Andreo at the start of this manual (“There are as many schools of astroimage processing as there are astrophotographers.”), astrophotographical image processing is more about interpretation of your data than presenting cold hard fact – there is no absolute 'right' way of processing an image and no one has a monopoly on asserting what constitutes a good image.

[2] http://www.global-rent-a-scope.com/
[3] http://www.lightbuckets.com/

Of course, that does not mean that best practices and techniques should be abandoned, but these words of wisdom are something to keep in mind should doubt set in.

Making your data StarTools friendly

There are some important things you should know about how modules work with pixel data in StarTools.

First of all, many algorithms require a mask (i.e. a second image which tells StarTools which pixels to operate on, and which pixels are off-limits) to operate effectively. As a matter of fact, a number of modules will not yield any useful results without a mask set. To find out more about masks, see the Masks chapter - page 150.

Secondly, the algorithms that StarTools employ are very sensitive to excessive noise, artificial artefacts (such as stacking artefacts, borders, text, etc.) and hot and/or dead pixels.

When processing images that contain such artefacts and noise, please be aware of their existence and try to eliminate them whenever possible before starting your processing. This may mean using the Crop tool to crop the image until stacking artefacts are no longer visible at the borders, or using the Mask, Heal and Layer modules to heal hot and/or dead pixels and/or noise (see examples on page 168 and 161).

Some modules come with an additional parameter that lets you specify a noisiness level of the image, so that the module can take precautions when processing noisy images. Failure to take noise and/or artefacts into consideration may lead to undesirable results and the introduction of further artefacts and/or other unexpected visual aberrations. In astronomical imaging, noise is the enemy and StarTools offers many ways of countering noise and prohibiting it from propagating by nipping it in the bud as early as possible.

What StarTools does and does not do

StarTools is an astronomical image post-processing tool - it takes raw stacked and aligned data and helps the user attain the best achievable result with that data.

StarTools, at present, does not perform stacking and/or aligning of images. There are many excellent utilities available that do this, some of them free of cost. Notable free software packages include Registax[4], Deep Sky Stacker (DSS)[5] and AVIStack[6] (all run on Windows).

StarTools, at present, does not offer any capturing or camera control abilities. For a low cost capturing solution for Windows and MacOSX that also performs basic stacking and basic image processing, please have a look at StarkLabs' excellent Nebulosity[7] software package. If you use a Canon DSLR on Windows, BackyardEOS[8] is another great capturing utility.

[4] http://www.astronomie.be/registax/

[5] http://deepskystacker.free.fr/english/index.html

[6] http://www.avistack.de/

[7] http://www.stark-labs.com/nebulosity.html

[8] http://backyardeos.binaryrivers.com/

Interface

Navigation within StarTools generally takes place between the main screen and the different modules. StarTools' navigation was written to provide a fast, predictable and consistent work flow. There are no windows that overlap, obscure or clutter the screen. Where possible, feedback and responsiveness will be immediate. Many modules in StarTools offer on-the-spot background processing, yielding quick final results for evaluation and further tweaking.

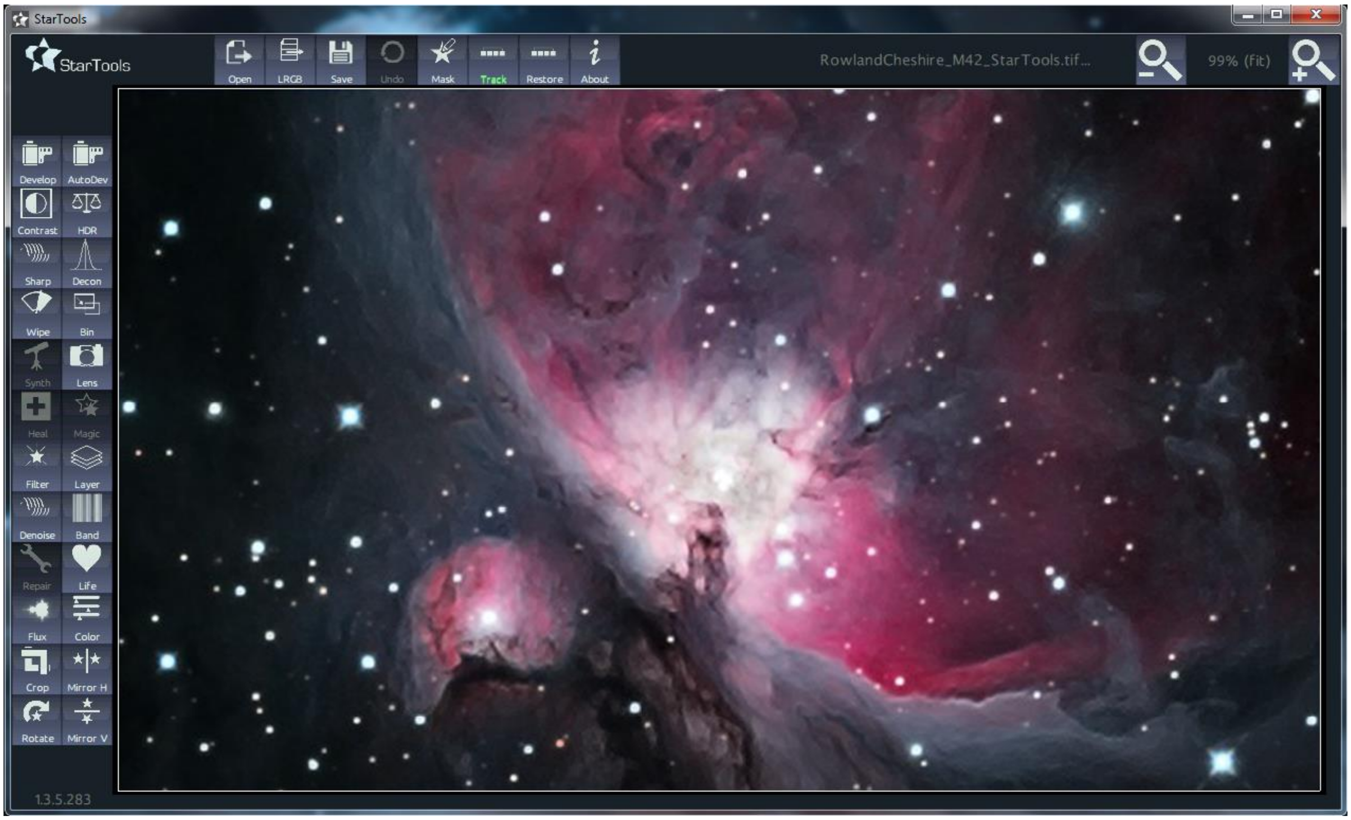


Illustration 1- The main screen interface (M42 courtesy of Rowland Cheshire)

In both the main screen and the different modules, a toolbar is found at the very top, with buttons that perform functionality that is specific to the active module. In case of the main screen, this toolbar contains buttons for opening an image, saving an image, undoing/redoing the last operation, invoking the mask editor and opening an 'about' dialog.

Exclusive to the main screen, the buttons that activate the different modules, reside on the left hand side of the main screen. Note that the modules will only successfully activate once an image has been loaded, with the exception of the 'LRGB' module (see page 26).

Consistent throughout StarTools, a set of zoom control buttons are found in the top right corner, along with a zoom percentage indicator.

Panning controls ('scrollbar style') are found below and to the right of the image, as appropriate, depending on whether the image at its current zoom level fits in the application window.

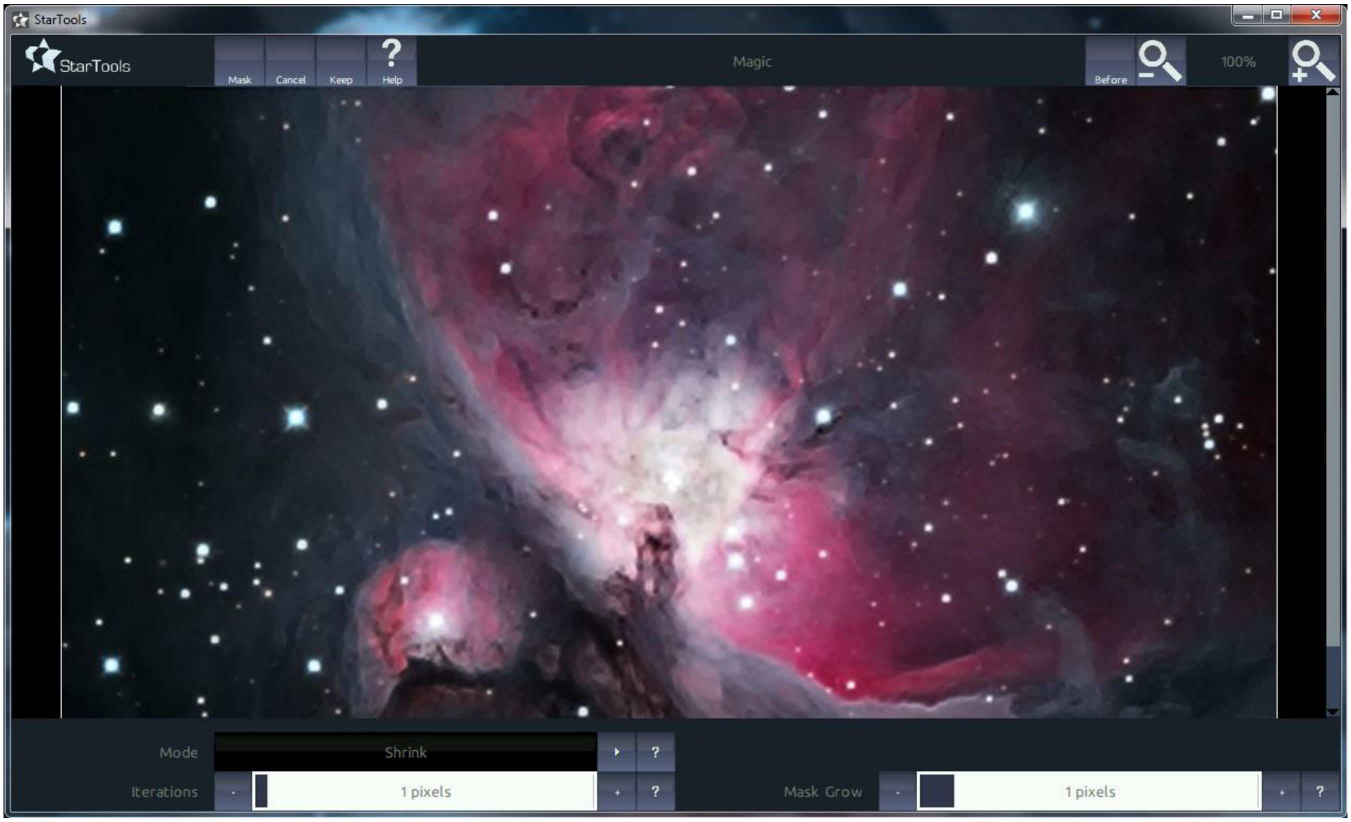


Illustration 2 – A typical module interface

The image in Illustration 2 shows the layout of a typical module (in this case the 'Magic' module), with the toolbar situated on top and the different parameters and settings, specific to the module, situated at the bottom.

Common to most modules is a 'Before/After' button, situated next to the zoom controls, which toggles between the original and processed version of an image for easy comparison.

All modules come with a 'Help' button in the toolbar, which explains, in brief, the purpose of the module. Furthermore, all settings and parameters come with their own individual 'Help' buttons, left to the parameter control. These help buttons explain, again in brief, the nature of the parameter or setting.

The parameters in the different modules are typically controlled by one of two types of controls

A level setter, which allows the user to quickly set the value of a parameter within a certain range.

An item selector, which allows the user to switch between different modes.

Illustration 3 – Level setter control

Setting the value represented in a level setter control is accomplished by clicking on the '+' and '-' buttons to increment or decrement the value respectively. Alternatively you can click anywhere in the area between the '-” and '+' button to set a value quickly.

Illustration 4 –Item selector control

Switching items in the item selector is accomplished by clicking the arrows at either end of the item description. Note that the arrows may disappear as the first or last item in a set of items is reached.

Track

Tracking plays a very large role in StarTools. This makes workflows much less linear and allows for StarTools' engine to "time travel" between different versions of the data as needed, so that it can insert modifications or consult the data in different points in time as needed ('change the past for a new present and future'). It's the primary reason why there is no difference between linear and non-linear data in StarTools, and you can do things in StarTools that would've otherwise been nonsensical (like deconvolution after stretching your data). If you're not familiar with Tracking and what it means for your images, signal fidelity and simplification of the workflow & UI, please do read up on it!

Tracking how you process your data also allows the noise reduction routines in StarTools to achieve superior results. By the time you get to your end result, the Tracking feature will have data-mined/pin-pointed exactly where the noise is in your image and apply noise reduction only in those areas. Therefore noise reduction is applied at the very end, as you switch Tracking off. This means that StarTools has had the longest possible amount of time to build and refine its knowledge of where the noise is in your image. This is different from other software, which allow you to reduce noise at any stage, since such software does not track signal evolution and its noise component.

Tracking how you processed your data also allows the color module to calculate and reverse how the stretching of the luminance information has distorted the color information (such as hue and saturation) in your image, without having to resort to 'hacks'. Due to this capability, color calibration is best done at the end as well, before switching Tracking off. This too is different from other software, which wants you to do your color calibration before doing any stretching, since it cannot deal with color correction after the signal has been non-linearly transformed like StarTools can.

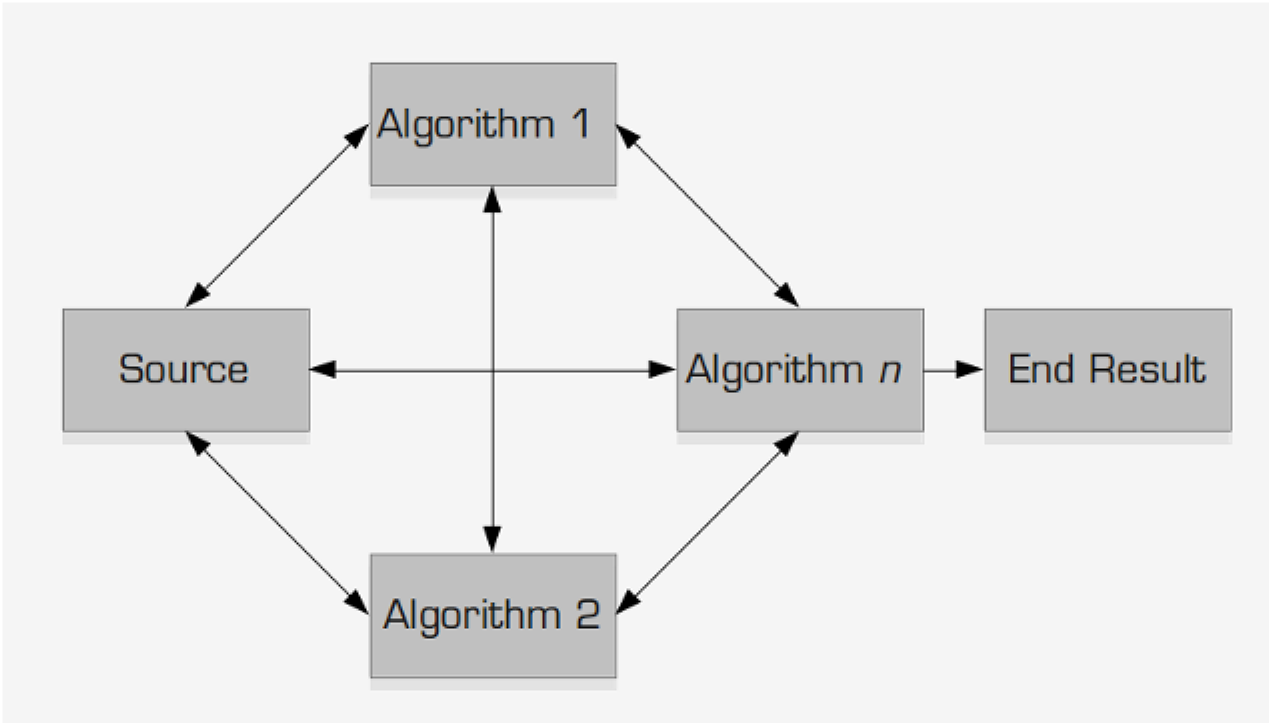
Tracking Details

As it stands today, the post-processing results you are able to achieve with StarTools are better than anything else on the market today (that we know of). Why do we feel totally comfortable making such a bold claim? Let us explain;

As opposed to all other software that is currently on the market, StarTools takes a completely different approach to the back-end processing under the hood.Rather than the traditional approach of having an application consisting of many granular algorithms, steps and filters that are not aware of each other (with one vendor even mistakenly touting an object-oriented approach as a 'feature'), StarTools is akin to a living and breathing organism; everything is connected and has its place, forming a whole that is greater than the sum of its parts. A module's influence or functionality does not stop when you close it. Its code is ever present, monitoring what you are doing and what other modules are doing. It is ever gathering data on what *you* are doing.

Traditional image processing software flow. Each algorithm/filter only has access to data as it was generated by the step immediately preceding it and only outputs data once to the step coming after it.

All modules that are accessible during StarTools' 'tracking' mode talk to each other and are aware of what all the other modules have done to your signal prior; StarTools observes how you stretch your signal and meticulously keeps track of noise propagation, levels and processing sequences throughout the processing. It does all of this in the background, and without bothering you (unless it really needs your input in matters that cannot be objectively determined - matters of taste and aesthetics).



StarTools interconnected processing. Each algorithm has access to the output of every other algorithm, independent of sequence, and each algorithm can send output retroactively (e.g. feedback) to any of the algorithms that were used any time to get to the result you are viewing at the moment.

It is this meta-data that StarTools collects that allows it to attain superior results on your behalf. For example, using this metadata, StarTools is able to perform effortless, extremely targeted noise reduction (it infers from your actions where the noise is), or hitherto impossible feats like mathematically-correct deconvolution of stretched data (it knows exactly how to temporarily undo any stretching you have done, and it knows exactly how to redo it as well with your newly deconvolved data). StarTools completely abstracts away the difference between linear and non-linear data and just seems to 'know' what to do with your data, it 'knows' which settings are required for best results and 'knows' how to derive individual (hidden) parameters from more abstract notions/descriptions and settings that make sense to human beings. You no longer have to tell the software these things - it has already inferred it. It's not magic, it's not black box voodoo - you are fully in control at all times. StarTools is simply being smarter about how your input and actions are relayed to all the different modules.

Modules

StarTools consists out of a number of integrated modules that perform different processing functions. Roughly, there are three types of modules

Modules that perform real-time processing in the background, while the user is free to tweak the parameters.

Modules that perform non real-time processing, where the user initiates the rendering once the desired parameters are set. These modules often require intensive calculations and will display a progress bar until the new result is available.

Hybrid modules, where some parameters are rendered real-time and others only come into effect once the user initiates rendering, during which a progress bar is shown until the new result is available.

All modules typically have a 'Cancel' button, which returns to the main screen ignoring the processed image and retaining the image as it was before the module was invoked.

All modules typically have a 'Keep' button, which returns to the main screen, keeping the image as it was processed by the module.

Some modules have 'preset' buttons. Clicking these buttons quickly recall useful default settings. These buttons are distinguished by their icon; they carry the same icon as their parent module in the main screen.

Some modules have a 'Mask' button, which invokes the mask editor. Masking is a very powerful feature and pivotal to using most modules effectively -greatly expands the amount of tools at your disposal. It is highly recommended that the user get acquainted with its operation.

Modules that have a Mask button will also blink the current active mask 3 times when the module is first invoked. This acts as a quick reminder of which pixels will be processed by the module, according to the mask set. For more information on how to use masks effectively, please see the Masks chapter - page 150.

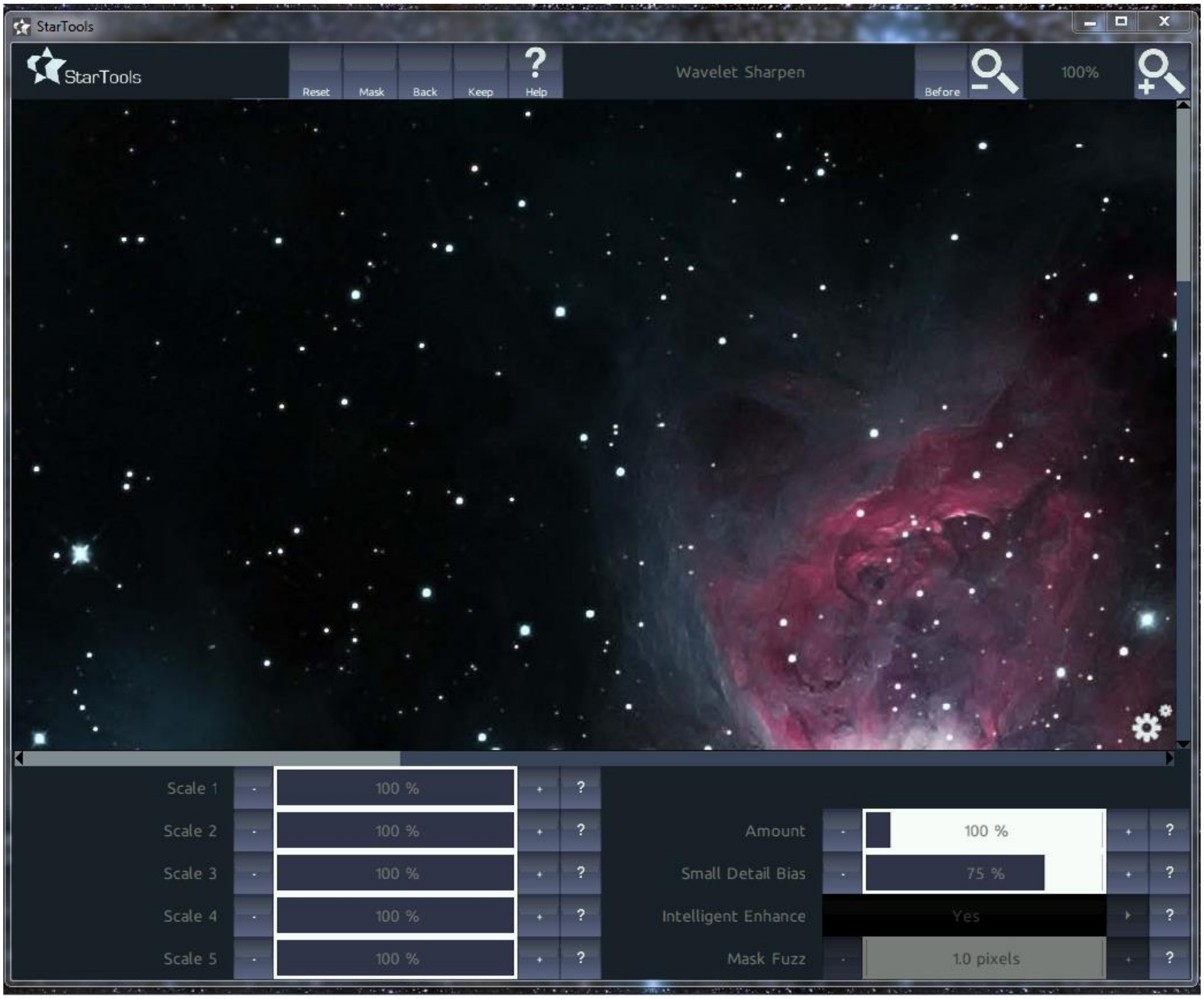


Illustration 5 – The Sharp module performing background processing

Modules that perform background processing while the user remains free to tweak parameters, will display a 'processing' icon in the form of two cogs in the lower right corner of the image, for example as seen in Illustration 5. This icon signifies that background processing is being performed and the image will be updated shortly with the new settings taking effect.

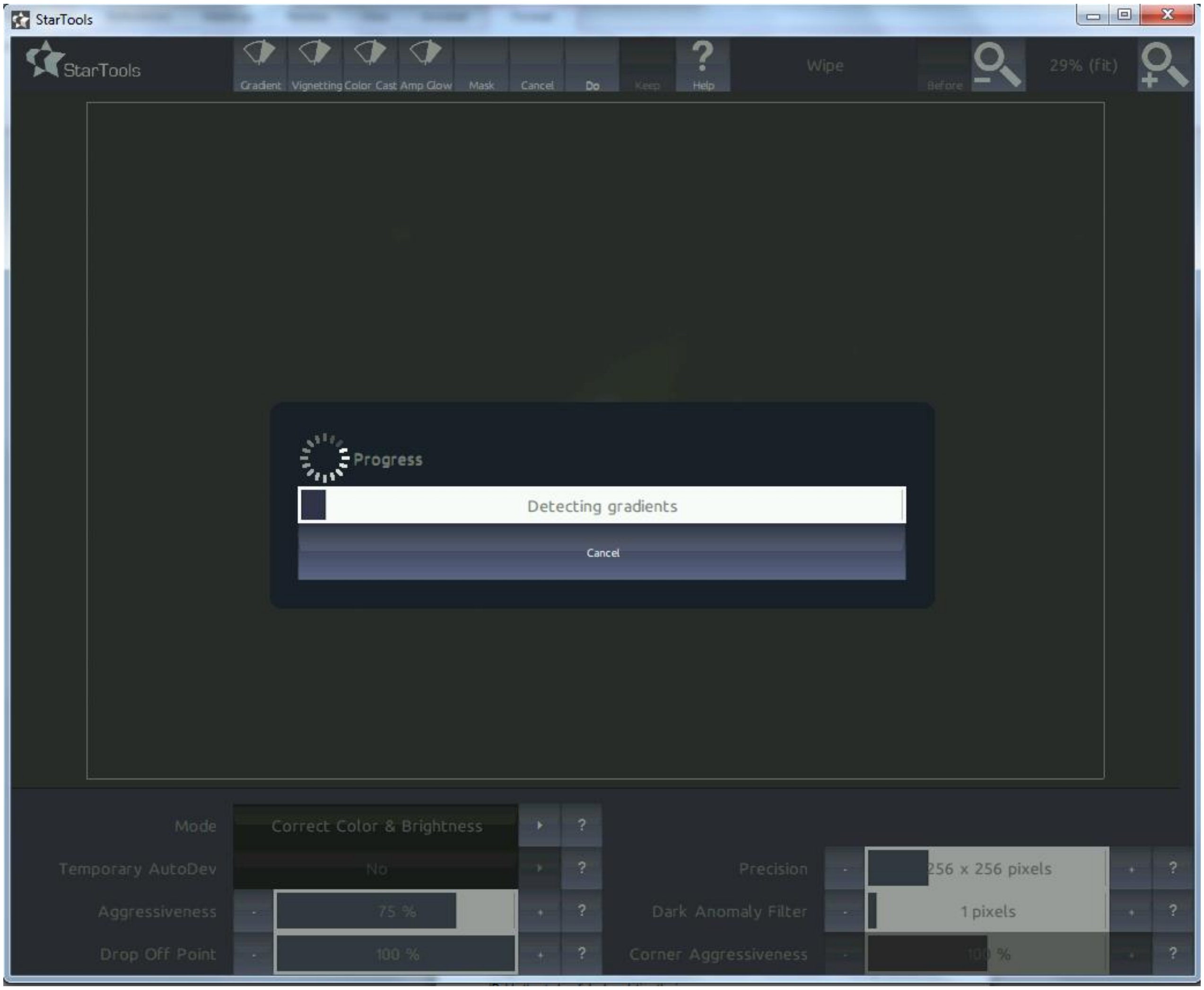


Illustration 6 – The Wipe module in the process of gradient modeling

Some modules will not automatically perform real-time processing in the background, due to the CPU intensiveness of their algorithms. These modules will have a 'Do' button. This button will initiate the processing once the user is satisfied with all parameters and settings. A status bar such as the one in Illustration 6 will show progress. Once the process completes, the image will update with the user may evaluate the result. Any subsequent tweaking requires the 'Do' button to be clicked, updating the image.

Open

Overview

StarTools reads FITS, PNG files and uncompressed, IBM byte-order TIFF files.

Stacking multiple frames is a virtual necessity in astrophotography, while calibration of your data with flats, darks and /or bias frames is highly recommended for best results. The FITS, uncompressed TIFF and PNG file formats are the file formats that such pre-processing software applications output. The reason for this is that, unlike lossy and 8-bit only formats (such as JPEG), the FITS and TIFF formats allow for storage of the high signal fidelity and precision that StarTools needs to get the most out of your image data.

JPEG files are especially unsuitable for astrophotography as it uses a lossy compression scheme that throws away data that StarTools needs. To make matters worse, JPEG files are stretched on-board and are often modified by the camera's on-board processing engine to appear sharper, and noise-reduced. This further interferes with the amount of detail that StarTools can recover from your data.

Usage

Click OPEN to browse and select a supported image file type.

StarTools will prompt for the 'Type of data' as follows:

- Linear, was not Bayered or is whitebalanced

  Use this option if your image is not from a one-shot color (OSC) camera or DSLR or if the data has already been white balanced (color corrected).

- Linear, was Bayered, is not whitebalanced

  If you are certain the image has not been color balanced and has indeed been debayered (which for a DSLR is indeed the case), then choose the 'Linear, was Bayered, is not whitebalanced' option. This will give you a small boost in noise reduction, owing to the fact that a bayer filter makes the CCD collect twice as many green pixels as red and blue pixels. StarTools can therefore assume that the green

channel has 2x better fidelity when considering its contribution to the luminance data.
However, if you have white balanced the data before handing it to StarTools, the channels will have been multiplied with some (to StarTools) unknown multiplier and thus this 2:1 ratio is no longer valid.

NOTE: This option assumes your OSC (or DSLR) uses a standard Bayer matrix color filter array (with one red, two green and one blue pixel for every cluster of four pixels), and you are sure the white balance was not in any way modified from what was recorded (by the camera or the stacking software).
NOTE: It appears that most DeepSkyStacker (DSS) data is color balanced (even though the user can tell DSS to expressly NOT color balance the data - it just gets ignored).

- Modified and linear

Modified and linear offers to attempt to reverse any stretch on the data:

Reverse stretch?

You have indicated that the data in this image has been stretched prior. If the data came directly from a JPEG or MPEG source (such as a digital camera or webcam) or is a stack of multiple frames from such a source, then StarTools can attempt to reverse this stretch. This may make it possible to avail of some of the benefits of StarTools' tracking mode, although your mileage may vary and linear data is highly recommended over this approach.

There are two choices to load the image data:

- Attempt to reverse stretch, activate tracking
- Don't activate tracking

- I’m not sure

  ‘I’m not sure’ displays the following:

> How can I tell if my DSLR data is linear, unstretched, non-whitebalance..
>
> You can tell if your data is linear and unstretched by looking at the brightness of the image. Only the brightest stars should be visible, while any DSO should be invisible or (if it has a very bright core) just have its core visible. Under no circumstance should you be able to see any background and under no circumstance should you be able to see any background noise.
>
> You will most likely see a strong background level that is much brighter than black. This background level should be colored, often a teal-like color. A brownish color indicates white-balance light pollution and thus means your image has been pre-white balanced.
>
> If you barely see any dominant color in your data, even after a quick initial stretch using the AutoDev or Develop modules, then chances are also that your data was white balanced or color calibrated.
>
> If you think your data is possibly stretched or whitebalanced, check the settings in your stacking software or consult the StarTools forums – we will likely be able to tell you definitively, and possibly also how to remedy this.
>
> Lastly, if you use Deep Sky Stacker to stack your images, have it output FITS files (not TIFF) and try using the Autosave.fits file if you find that ‘regular’ saving outputs a stretched or modified image.

LRGB

Overview

The LRGB module is a flexible colour image compositor. It allows you to mix and match Luminance, Red, Green and/or Blue images.

LRGB features automatic colour interpolation to make up for any missing channels. This feature is particularly useful when, for example, creating a composite from Ha and Hb data in the red and blue channels. LRGB will automatically generate the green channel in this instance.

LRGB also imports colour images and extracts channels as appropriate. For example, importing a colour image into the red channel, will automatically only extract the red data from the colour image.

Please note, that when importing files, all files must have the same dimensions. There is one exception: red, green or blue channels may be exactly ½ the size of the luminance channel. This is so that the luminance channel can be recorded at 1x1 hardware binning, while red, green and blue may be recorded at 2x2 hardware binning. In this case the red, green and blue files will be automatically scaled up to match the size of the luminance channel.

Usage

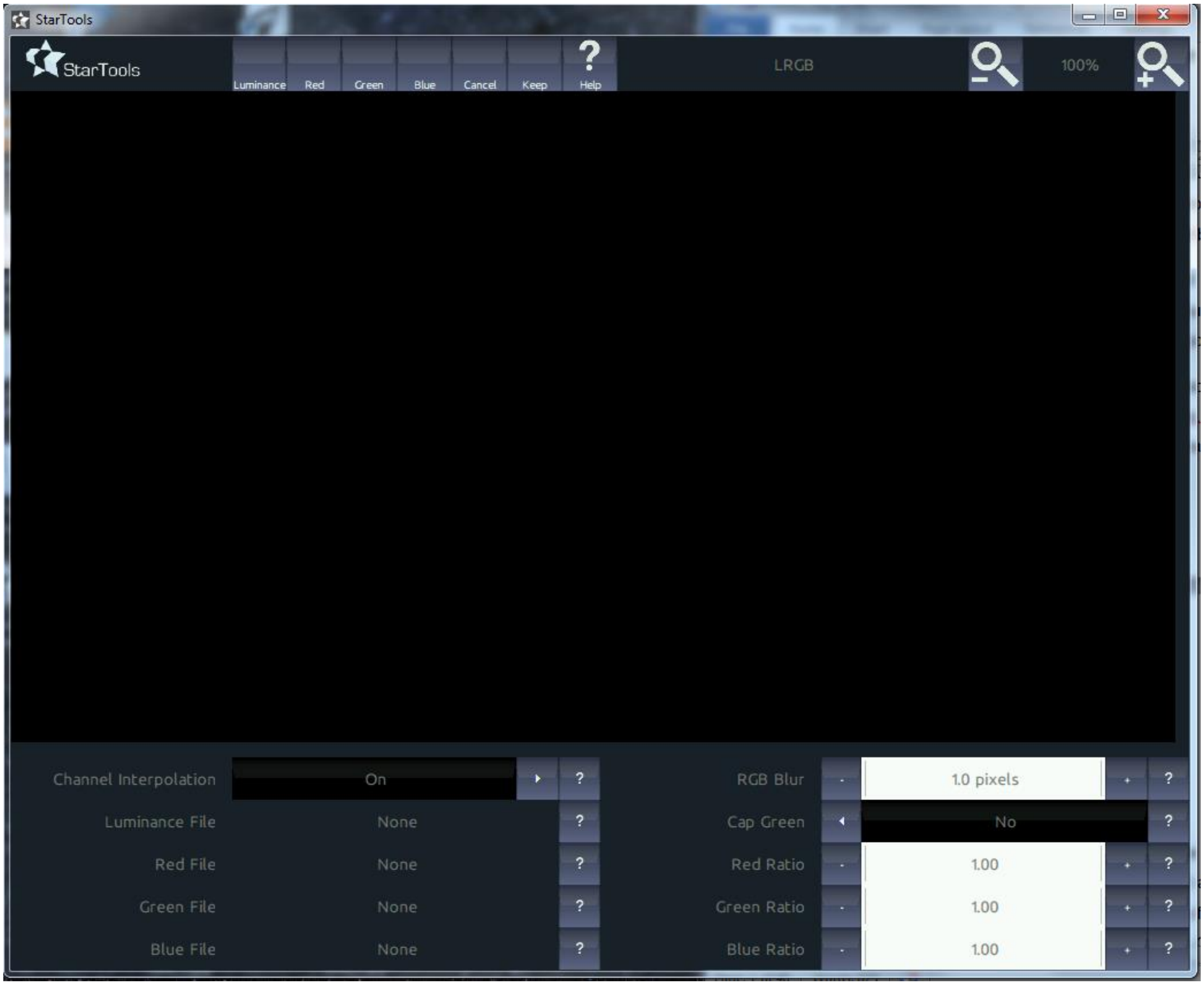


Illustration 7 – LRGB module interface

The LRGB module has 10 adjustable parameters and settings

- RGB Blur - specifies the kernel radius of a Gaussian blur that is applied to the red, green and blue channel in case of importing an LRGB image. Note that this option only takes effect if a luminance channel is present. Often more imaging time is spent on acquiring the luminance frame, to which the human eye is much more sensitive. The red, green and blue data can often be of a much lower quality (and thus noisier) without seeing much difference. Blurring the red, green and blue data effectively acts as a low-pass filter, eliminating the noise in the red, green and blue channels. Even with heavy blurring of the red, green and blue channels in an LRGB composite, the difference may be negligible, however colour noise will be all but eradicated.

- Channel interpolation - toggles the interpolation of missing channels on or off. This feature is particularly useful when, for example, creating a composite from Ha and Hb data in the red and blue channels, without a green channel. LRGB will automatically generate the green channel in this instance. This feature is not just limited to the green channel - It will do the same for any other missing channel.
- Cap green - specifies whether LRGB should block any green hues from forming in the image. The reasoning is that very few, if any, objects in outer space emit dominant green light.
- Red Ratio - adjusts a multiplier that should be applied to the red channel. Note that top-end clipping may occur as a result. This value may be used to attain the exposure duration equivalent of the other channels. For example, if the red frame was a 30 minute exposure and the green and blue frames were both 45 minute exposures, then the red ratio should be set to 1.5 to make up for the red channels' shorter exposure duration.
- Green Ratio - adjusts a multiplier that should be applied to the green channel. Note that top-end clipping may occur as a result. This value may be used to attain the exposure duration equivalent of the other channels.
- Blue Ratio - adjusts a multiplier that should be applied to the blue channel. Note that top-end clipping may occur as a result. This value may be used to attain the exposure duration equivalent of the other channels.
- Luminance File - shows the path of the file that is used as the luminance channel. Note: a colour file used for this purpose will be converted to greyscale by adding up all pixel values (no color space conversion is performed).
- Red File - shows the path of the file that is used as the red channel. Note that if this file is a colour file, then the red channel is extracted from this file.
- Green File -shows the path of the file that is used as the green channel. Note that if this file is a colour file, then the green channel is extracted from this file.

- Blue File -shows the path of the file that is used as the blue channel. Note that if this file is a colour file, then the blue channel is extracted from this file.

Develop

Overview

DEVELOP allows you to ‘develop’ an image like you would in a digital darkroom. It also allows you to adjust general brightness levels. You can also use it to quickly inspect your data (that would otherwise be too dark to see), or as a tool to perform final global stretching.

DEVELOP automates the process of finding the correct black point and white point in your image. This way the full dynamic range that your image requires is precisely pinpointed and allocated. This ensures your image will be able to attain maximum contrast as you process it further. Developing is easy as pulling the DIGITAL DEVELOPMENT and/or GAMMA slider along.

Drag the DARK ANOMALY slider along until you can’t see the image darken much. This will make sure the black point is set to the darkest pixel that constitutes ‘real’ data. Spurious dark pixels (such as dead pixels) will not be considered this way.

Together with the AUTODEV, CONTRAST, HDR and SHARP modules, StarTools aims to make tedious (and sub-optimal) manual curve manipulation and histogram checking obsolete. Modules in StarTools never clip your data, unless explicitly allowed to do so.

NOTE: When stretching in the DEVELOP module, it is only important that you bring out the faintest details you want visible. Subsequent application of the CONTRAST, HDR and SHARP modules will assist you in toning down parts of the image as required. They will also assist you in revealing detail in the very brightest parts of the image.

If StarTools ‘tracking’ feature is enabled and if you launch the DEVELOP (or AUTO DEVELOP) module again after stretching a first time, you have the option of going back in time and redoing (replacing) that stretch. In that case, any operations you performed after that initial stretch that you are about to redo will be automatically redone for you.

IMPORTANT: If you are not merely using this module for a ‘quick stretch’ to inspect your data, then remove any stacking artifacts by using the CROP module before using this module for final stretching. After doing this simply revisit this (or the AUTO DEV) module for the final stretching.

Stacking artifacts interfere with the automatic dynamic range optimization process. In the presence of dead pixels or small dust donuts, increase the DARK ANOMALY FILTER parameter.

Usage

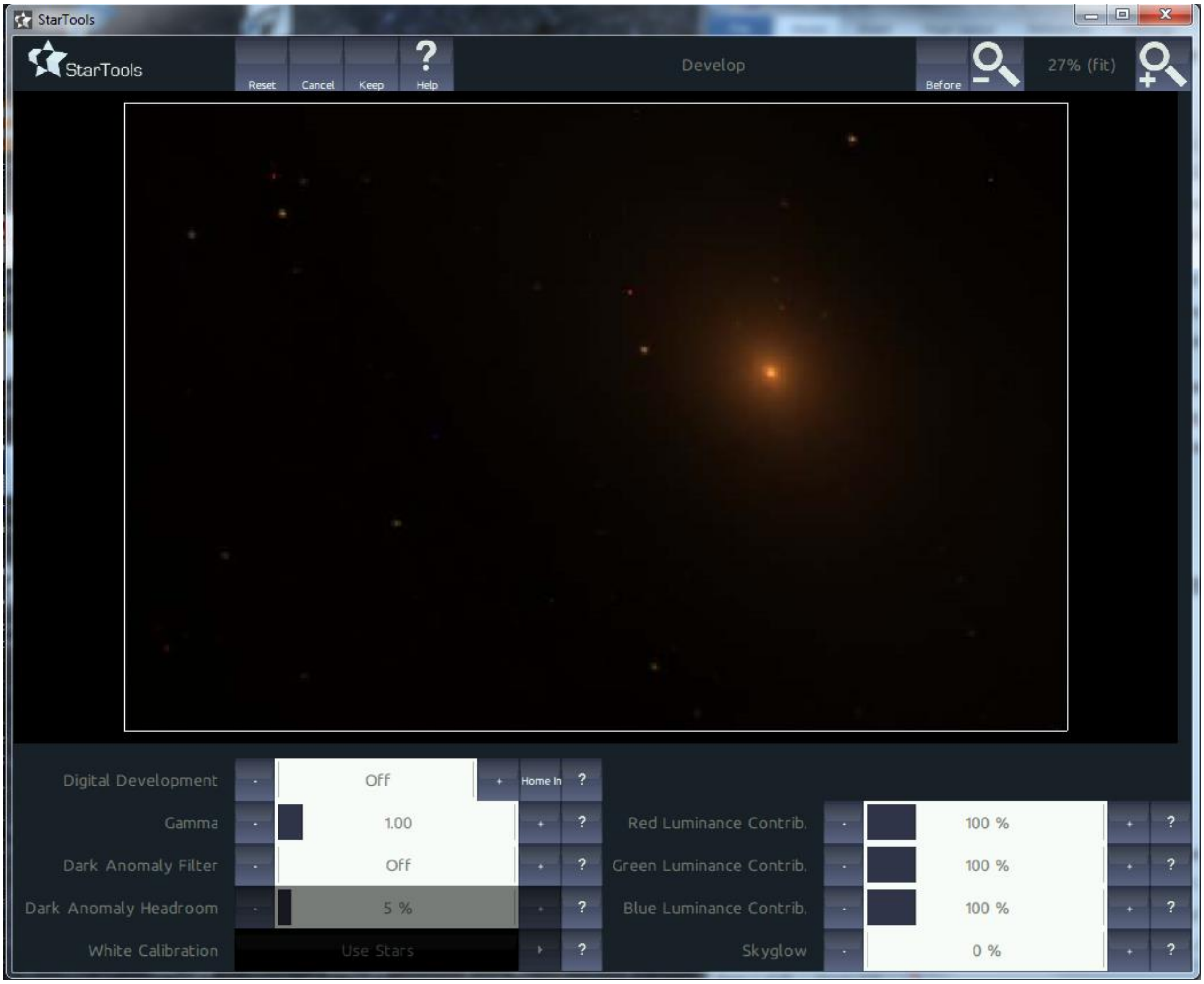


Illustration 8 – The Develop module screen

- [DIGITAL DEVELOPMENT] emulates the typical exposure response of film. If using the DEVELOP module for stretching your data for further processing, just drag the slider along until you feel that the faintest parts of the image are visible enough and there is good balance between bright and dark. Don't worry about making detail in bright cores too bright - you will be able to recover this detail with the CONTRAST and HDR modules.

The HOME IN button next to the slider makes it even easier to find the right balance. Simply click on it repeatedly until it settles on a god value for your image.

- [GAMMA] performs a gamma adjust of the image with the specified power factor. It is the simplest non-linear stretch available. Most computer screens require a gamma of 2.2 for correct conversion of linear data into non-linear data that appears linear on-screen.

- [DARK ANOMOLY FILTER] specifies how strong the DEVELOP module should filter the image in order to get rid of any 'dark anomalies'. Anomalies that are darker ('dark anomalies') than the galactic background (such as dead pixels, stacking artifacts, dust particles on the CCD, or terrestrial objects such as trees, mountains, etc) may throw the DEVELOP module's automatic background level detection off in some cases. The [DARK ANOMALY FILTER] can remove small dark anomalies such as dead pixels, small dust specks or small scratches. For larger anomalies, please use the HEAL or CROP modules to heal or crop out the anomalies.

- [DARK ANOMOLY HEADROOM] specifies what to do with dark anomalies found by the DARK ANOMOLY FILTER. This parameter specifies how much headroom is re-allocated for them. More specifically, it specifies how many gray levels they may take up as a percentage of the total amount of gray levels they used to occupy.

  NOTE: this parameter only takes effect when [DARK ANOMOLY FILTER] is not 'off'. It is grayed out otherwise.

- [WHITE CALIBRATION] specifies how the DEVELOP module should detect the white point in the image (e.g. the brightest level in the whole image).

  The easiest way for the DEVELOP module to detect the correct white point is by looking at the presence of big bright stars that have white cores – i.e. stars that have a completely white (overexposed) core. This way of calibrating the white point is activated by choosing the USE STARS setting, which is also the default.

  However, in rare cases where no bright stars are present and the brightest features are hot (defective) pixels or noise, then the DEVELOP module can optionally use the DARK ANOMALY FILTER to also filter out 'white' anomalies, just as it filters out the 'dark' anomalies. This alternative way of calibrating the white point is activated by choosing the USE DARK ANOMALY FILTER (May White Clip)] setting. As indicated by 'May White Clip' between the parentheses the latter setting may cause some white clipping in the form of star cores being blown out. This can be rectified at a later stage by blending non-blown out cores back in using the LAYER module.

[RED LUMINANCE CONTRIBUTION] adjusts the contribution of the red channel to the image's overall luminance. If your red channel's data is less reliable (for example due to lack of an IR filter or due to chromatic aberration), reduce this value. This parameter has no effect on black and white images (since there is no red channel) and is therefore disabled when processing such images.

Illustration 9 – Raw Jupiter details with all channels Contributing to luminance

Illustration 10 – Raw Jupiter details with only the green channel contributing to luminance

- [GREEN LUMINANCE CONTRIBUTION] adjusts the contribution of the green channel to the images over luminance. If your green channel's data is less reliable (for example due to chromatic aberration), reduce this value. This parameter has no effect on black and white images and is therefore disabled when processing such images.

- [BLUE LUMINANCE CONTRIBUTION] adjusts the contribution of the blue channel to the images over luminance. If your blue channel's data is less reliable (for example due to chromatic aberration), reduce this value. This parameter has no effect on black and white images and is therefore disabled when processing such images.

- [SKY GLOW] reintroduces an artificial sky glow. The background in images taken from earth is typically not perfectly black, but a slight gray (or blue), due to various factors such as atmospheric scattering, Genenschein, etc.

  It is best to use this feature as one of the last steps when finishing an image, rather than during initial stretching, just before saving your image.

AutoDev

Overview

AUTO DEVELOP is the quickest and most powerful way in StarTools to globally stretch your images for inspection and further processing. All you have to do is tell it the size of the features it can safely ignore (such as single pixel noise grain) and it will do its best to give the features that are interesting in your image as much dynamic range as possible, throughout your image.

You can even specify an area (‘region of interest’) of the image where you think the most interesting objects are located in you mage. AUTO DEVELOP will then give priority to bringing out objects in that area specifically, as opposed to bringing out objects in the image as a whole.

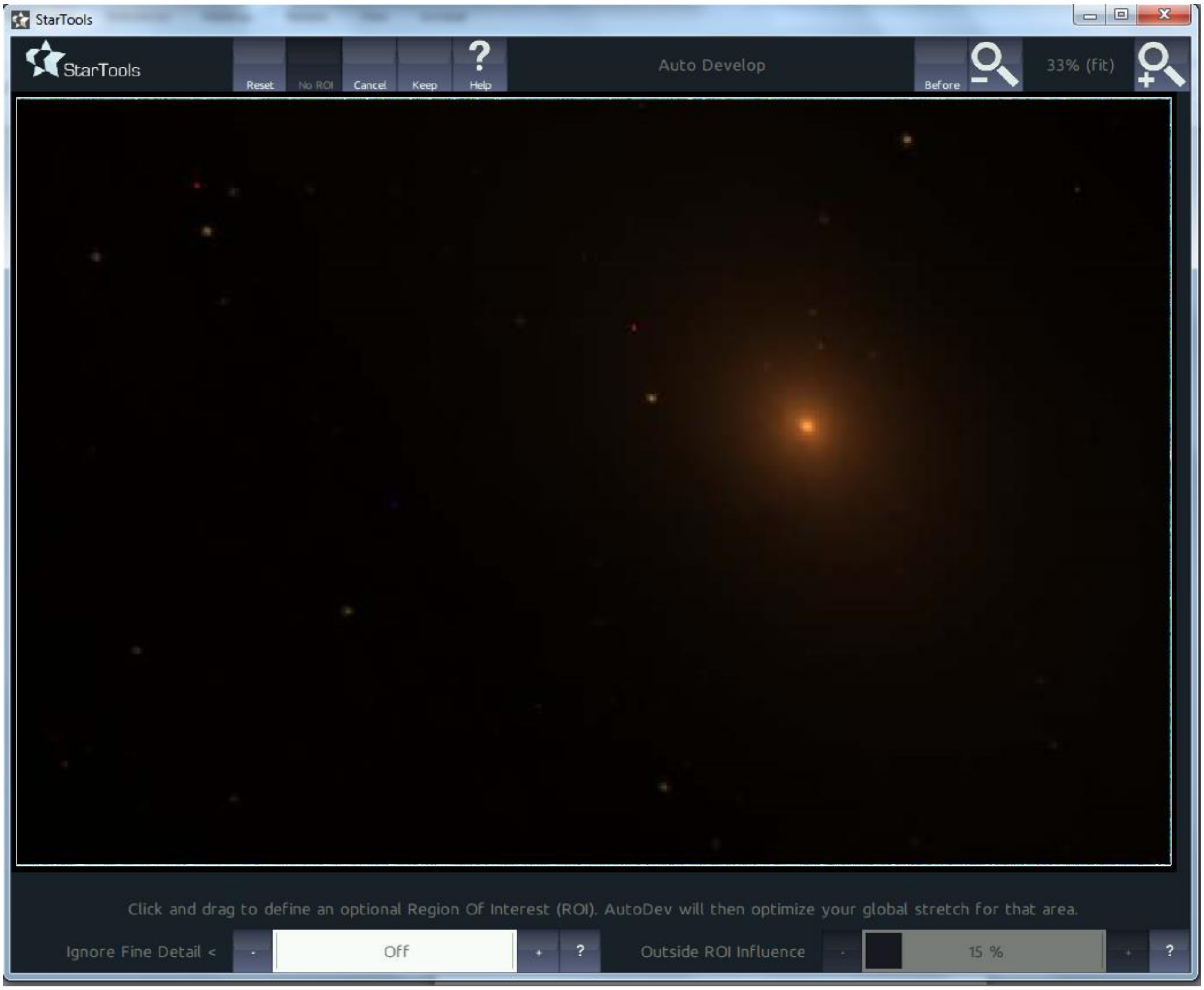


Illustration 11 – The AutoDevelop module interface screen

If StarTools 'tracking' feature is enabled and if you launch the AUTO DEVELOP (or DEVELOP) module again after stretching a first time, you have the option of going back in time and redoing (replacing) that stretch. In that case, any operations you performed after that initial stretch that you are about to redo will be automatically redone for you.

Because AUTO DEVELOP is designed to bring out as much detail as it can, it is a great tool to highlight any problems in your image; for example, if the image has a very strong color bias, you are probably seeing the effects of light pollution (you'll want to correct this using the WIPE module). Other problems that AUTO DEVELOP will highlight are noise, vignetting and lack of flat frames (ex. Dust donuts).

AUTO DEVELOP achieves best results on data that has had proper flat and dark frames applied. In the absence of such data, data may be corrected using, for example the WIPE module, after which the AUTO DEVELOP module can be reapplied using the corrected data.

AUTO DEVELOP never clips your data (unless an explicit Region of Interest is used and OUTSIDE ROI INFLUENCE is set to 0%) and never blows out star cores.

IMPORTANT: If you are not merely using this module for a ‘quick stretch’ to inspect your data, then remove any stacking artifacts by using the CROP module before using this module for final stretching. After doing this, simply revisit this (or the DEVELOP) module for the final stretching. Stacking artifacts interfere with the automatic dynamic range optimization process.

Contrast

Overview

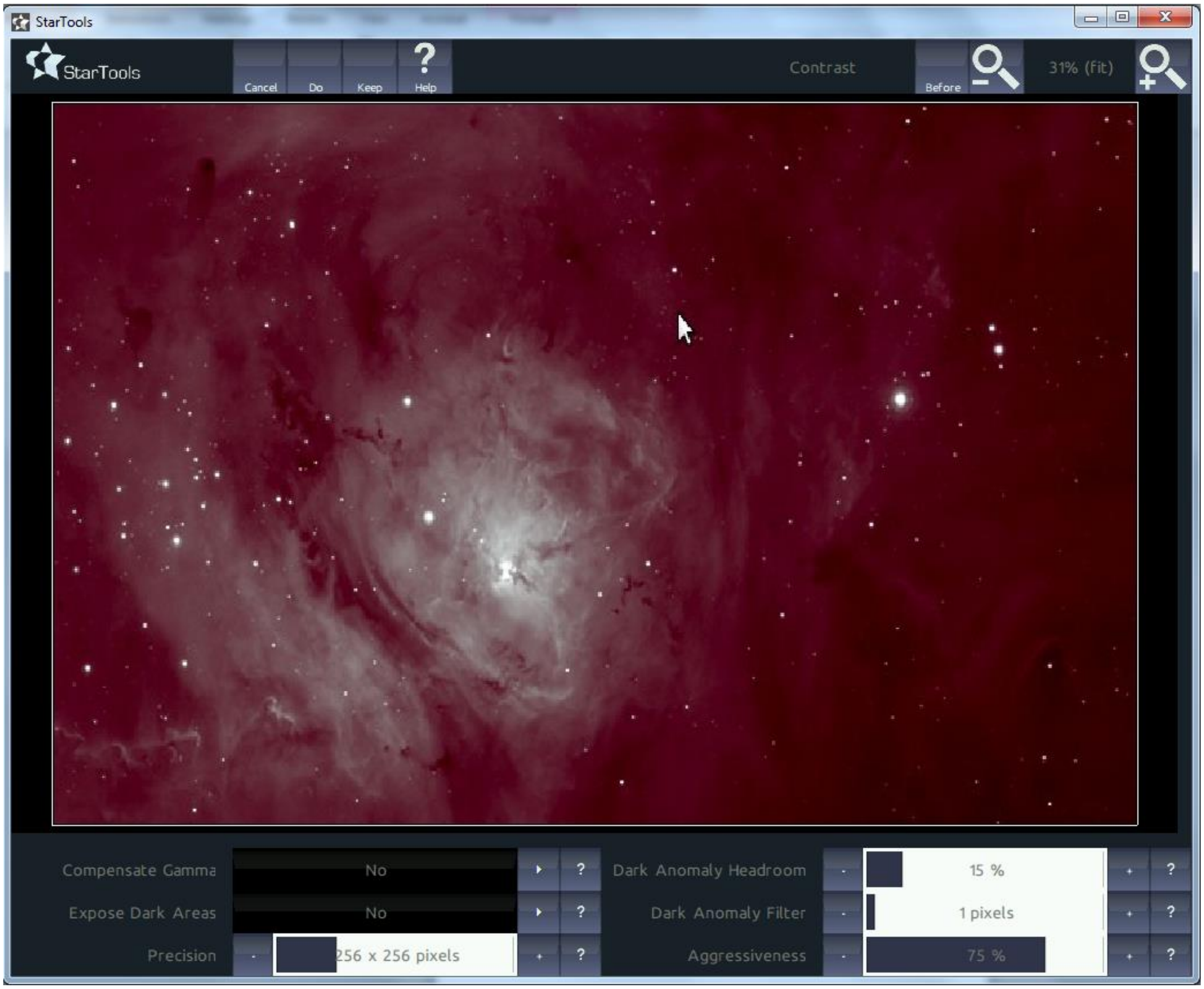


Illustration 12 – The Contrast module 1st screen interface

The CONTRAST module increases local contrast in your image and is an excellent tool to follow a global stretch that was performed using the DEVELOP or AUTODEV modules. The CONTRAST module usually does a great job unaided, however dark anomalies in an image such as dead pixels, dust particles, or terrestrial objects such as trees, mountains, etc, may throw CONTRAST's local brightness level detection off. If anomalies are small (e.g. hot or dead pixels), you may want to use the DARK ANOMALY FILTER parameter.

NOTE: If you do not take into account hot and dead pixels, artifacts in the form of halos may arise in some cases.

Usage

- [COMPENSATE GAMMA] specifies whether CONTRAST should compensate for the darkening of an image by increasing the gamma (i.e. performing a slight non-linear stretch).
- [EXPOSE DARK AREAS] specifies whether CONTRAST should be allowed to brighten certain parts of the image that benefits contrast.
- [PRECISION] specifies the amount of samples WIPE takes from the image. In large images with quickly changing gradients, higher precision may be appropriate.
- [DARK ANOMOLY HEADROOM] specifies what to do with dark anomalies found by the DARK ANOMOLY FILTER. This parameter specifies how much headroom is re-allocated for them. More specifically, it specifies how many gray levels they may take up as a percentage of the total amount of gray levels they used to occupy.

  NOTE: this parameter only takes effect when DARK ANOMOLY FILTER is not 'off'. It is grayed out otherwise.
- [DARK ANOMOLY FILTER] specifies how strong the DEVELOP module should filter the image in order to get rid of any 'dark anomalies'. Anomalies that are darker ('dark anomalies') than the galactic background (such as dead pixels, stacking artifacts, dust particles on the CCD, or terrestrial objects such as trees, mountains, etc) may throw the DEVELOP module's automatic background level detection off in some cases. The DARK ANOMALY FILTER can remove small dark anomalies such as dead pixels, small dust specks or small scratches. For larger anomalies, please use the HEAL or CROP modules to heal or crop out the anomalies.
- [AGGRESSIVENESS] specifies how aggressive WIPE should be when removing gradients. The higher the values, the more aggressive WIPE becomes when removing local gradients.

Please note that when optimising the contrast of wide field images, with lots of empty space, Contrast may introduce artefacts in the darker parts of the image, brightening some parts that should not be brightened. To avoid such artefacts, use the 'DSO Darken' mode.

Also, please note that, similar to the Wipe module, the Contrast module is susceptible to noise and may cause artefacts when presented with noisy images, stacking artefacts, dead pixels and/or hot pixels.

HDR

Overview

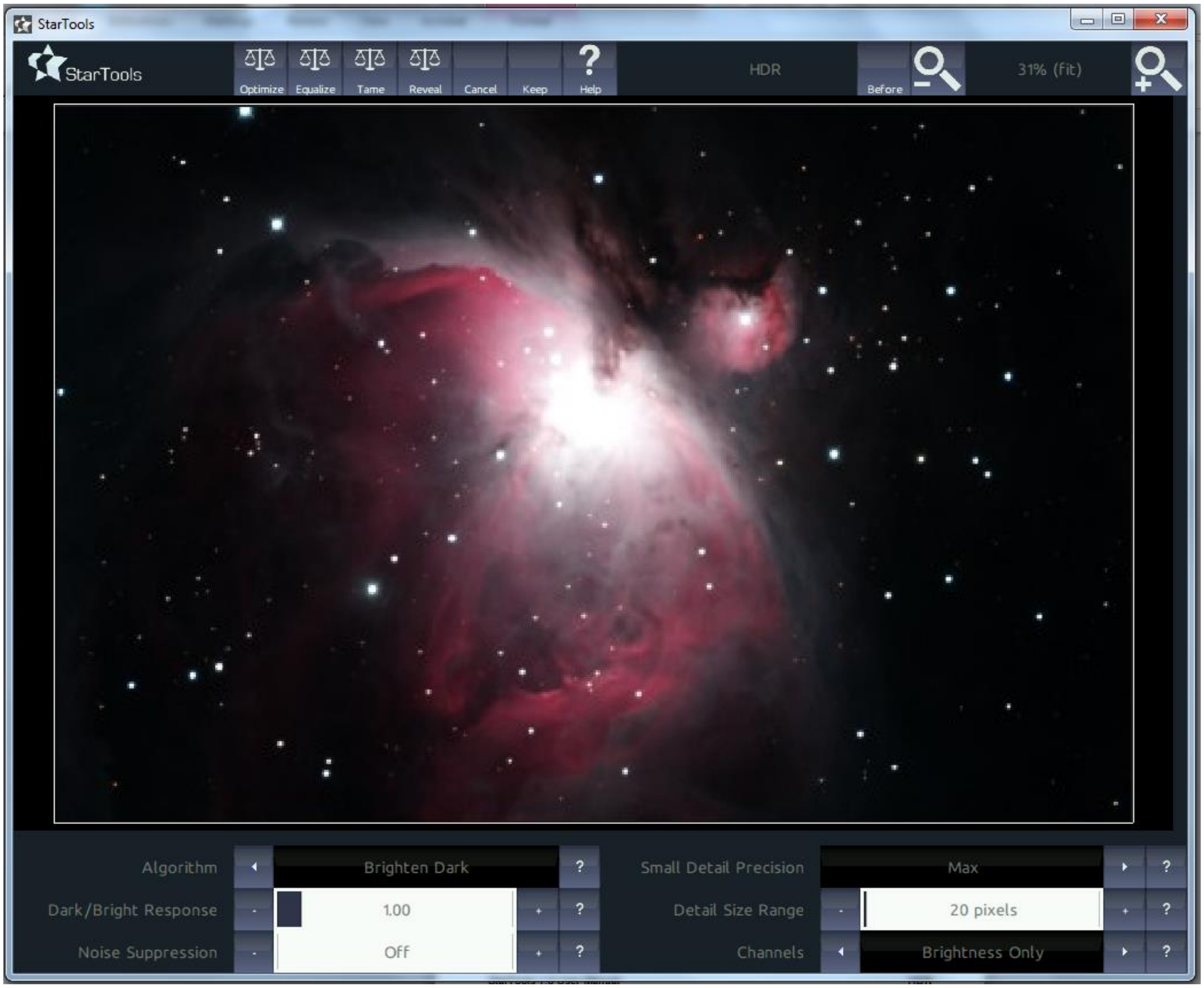


Illustration 13 – The HDR module screen interface

The HDR module optimises an image's dynamic range. Its algorithm was designed to bring out faint detail that is hidden in under and over-exposed parts of an image by locally increasing the dynamic range. The algorithm works by intelligently evaluating the image at different exposures and picking the optimal brightness for each pixel.

Illustration 14 – M42 image before HDR treatment (image courtesy of Rowland Cheshire)

Illustration 14 shows an image of M42 before HDR treatment.

Illustration 15 – M42 image after HDR treatment with the default parameters

Illustration 15 shows the same image of Illustration 14 after HDR treatment with all parameters at their default values. Notice how the brightness of the core has been toned down, while new detail has become visible. Also notice that stars that were already visible have not been affected, while some new stars have appeared. The HDR module has successfully manipulated the local dynamic range to bring out new detail.

Usage

[HDR] offers you some very powerful ways to optimize an images' dynamic range allocation locally, in near real-time. Together with the DEVELOP, AUTODEV and CONTRAST modules, it is an automated replacement for the manual curve manipulation. The HDR module typically yields better results than manual curve tweaking due to its local approach. The use of the HDR module is generally only applicable after stretching with the DEVELOP or AUTODEV modules.

The HDR module will help you dig out every last little bit of detail that is hidden in the highlights or shadows of your image. It accomplishes this in near real-time, allowing you to see the immediate effects of your chosen settings. HDR can operate on the brightness of image details, on the colour of image details or both.

The HDR module provides Optimize, Equalize, Tame and Reveal algorithm preset buttons.

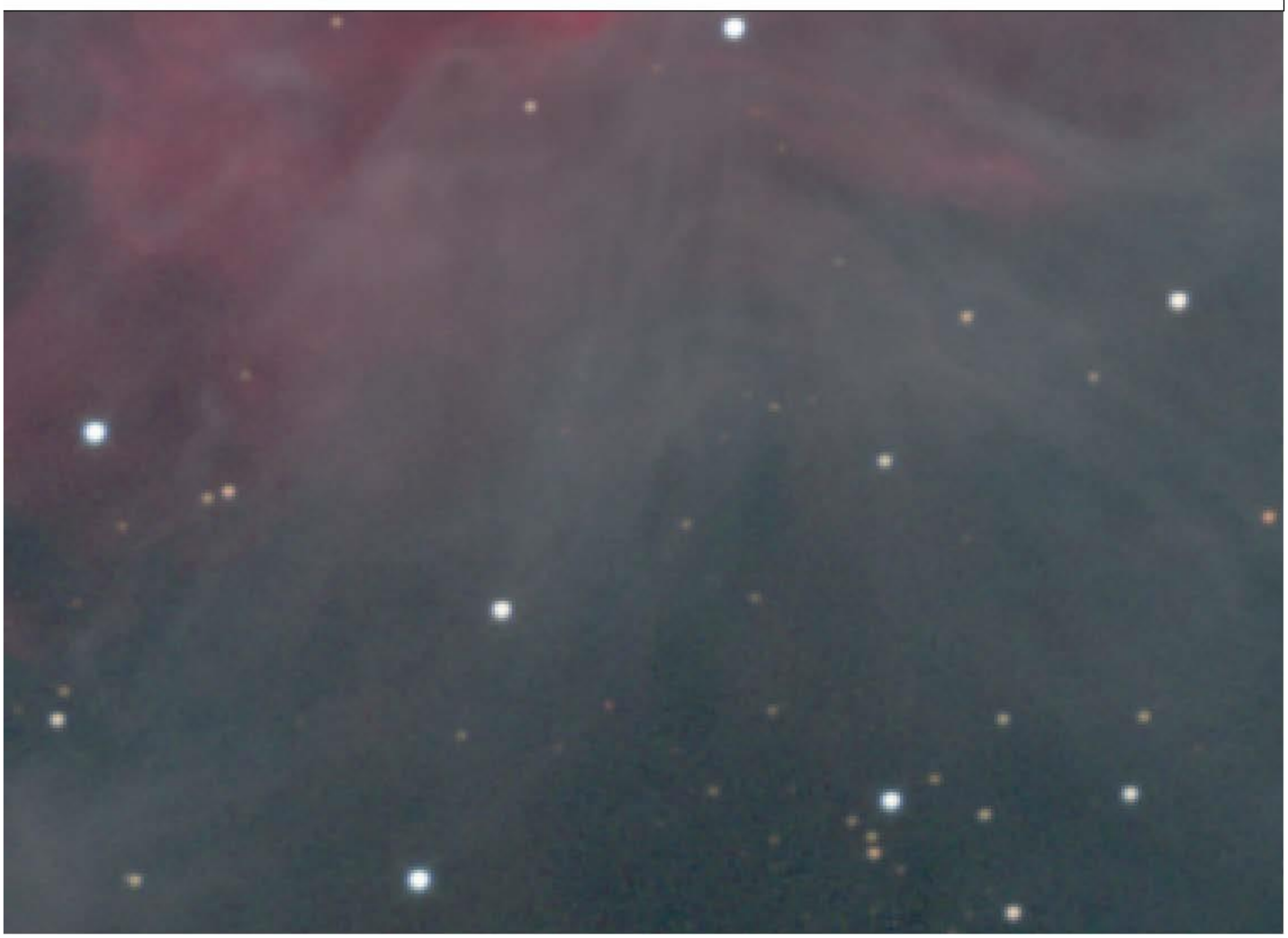


Illustration 16 – 3x magnifies crop of M42 with 'Noise Suppression set to 'off'

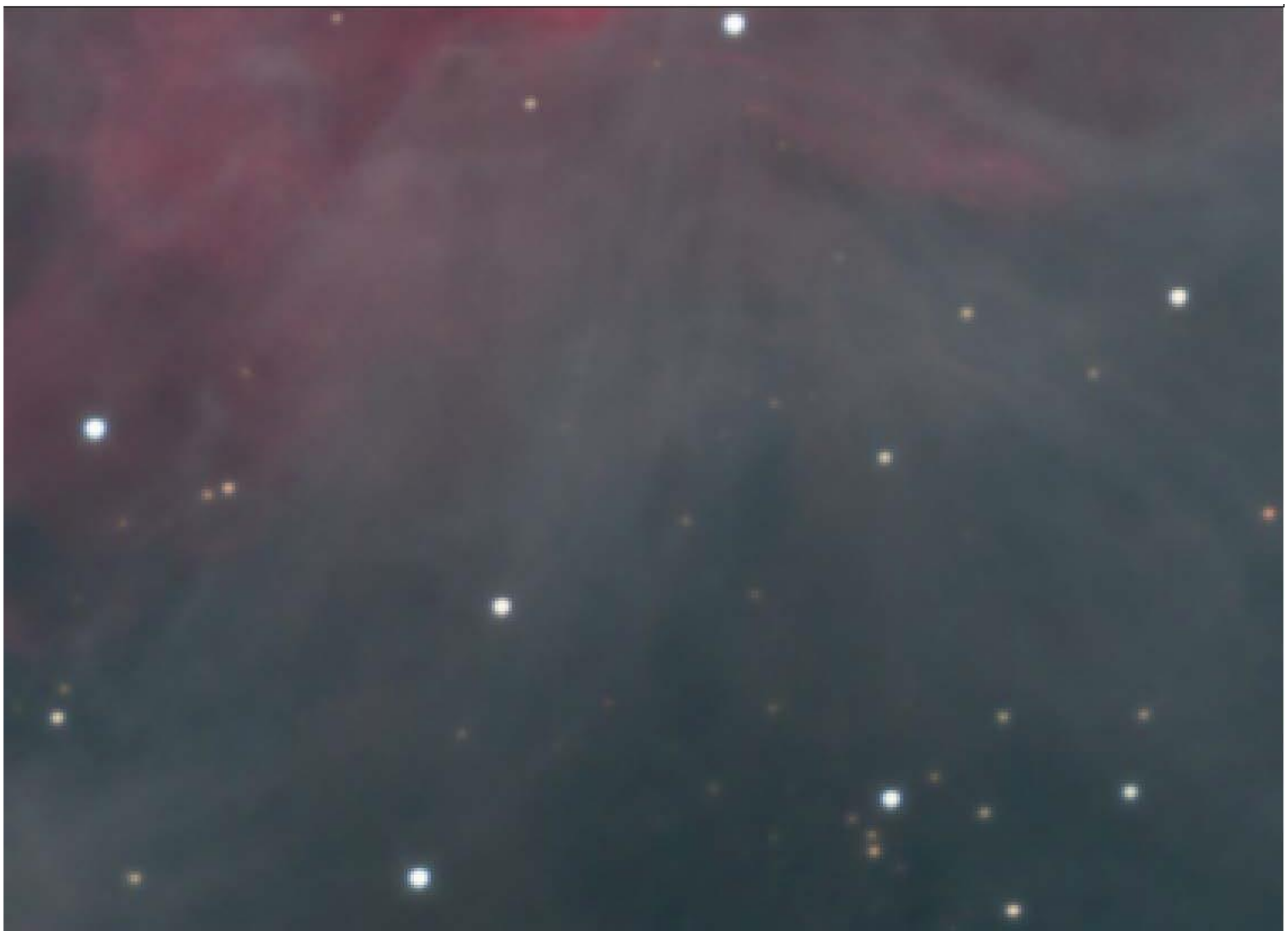


Illustration 17 – 3x magnified crop of M42 using 'Noise Suppression'

- [ALGORITHM] selects the main algorithm HDR should use.
    - [EQUALIZE] optimizes both shadows and highlights, so that any detail that is present in either is brought out.
    - [OPTIMIZE SOFT] accentuates detail it can find.
    - [OPTIMIZE HARD] is a more aggressive version of OPTIMIZE SOFT.
    - [TAME HIGHLIGHTS] TONES DOWN BRIGHT AREAS, SUCH AS BRIGHT CORES. It is similar to the EQUALIZE algorithm, only it doesn't brighten faint detail.

  - [REVEAL DSO CORE] is an extremely effective algorithm for revealing detail in almost-overexposed cores, such as the cores of M8, M42 and other Deep Space Objects, such as galaxies, that pose dynamic range challenges.

  - [BRIGHTEN DARK] brightens areas, digging out faint detail. It is similar to the EQUALIZE algorithm, only it doesn't tone down highlights.

- [DARK/BRIGHT RESPONSE] specifies how strong the effect of your chosen algorithm should be in the very brightest and darkest parts of your image.

- [NOISE SUPPRESSION] specifies how much noise suppression HDR should apply.

- [SMALL DETAIL PRECISION]
  - MAX
  - MEDIUM
  - LOW
- [DETAIL SIZE RANGE] [MOST AFFECTED DETAIL SIZE] specifies what detail size HDR should concentrate on, in order to bring out the most interesting detail. All detail brought out will be embedded in the context of this 'master' detail size. This ensures a natural looking image with enhance detail that sits well within the context of its parent super structure.

- [CHANNELS] specifies how whether HDR should enhance the recovered detail's brightness, color or both.

Sharp

Overview

The Sharp module offers some very flexible ways of sharpening an image.

Illustration 18 – Crop of M42 (image courtesy of Rowland Cheshire, unsharpened

Illustration 19 – Crop of M42, sharpened

A sharpening tool is an important tool in any astrophotographer's toolbox. It emphasises details in an image that would otherwise be hard to notice.

WAVELET SHARPEN offers a powerful way of sharpening your image by decomposing the image into multiple layers of different scale sizes. These layers can then be individually modified to achieve scale dependent sharpening.

StarTools' implementation of wavelet sharpening is set apart from other implementations by its detail preservation safe guards. These safe guards mean that it is impossible to clip (and destroy) your data and that an (optional) detail checking feature ensures that sharpening is only applied to parts of your image that actually stand to gain from it.

StarTools' Sharp module offers a selection of 6 algorithms to sharpen your image with, along with several other features that can be used in conjunction to improve your results even more.

When you think of sharpening, you typically think of revealing fine detail. While the Sharp module is certainly more than capable of this, the Sharp module goes much further - the Sharp module can be made 'scale aware'.

Scale aware sharpening allows you to control the size of the features that you wish to sharpen. For instance, you can now make the spiral arms in a galaxy stand out, without sharpening any further details that are smaller or larger. In some ways, this makes StarTools' Sharp module very similar to traditional Wavelet sharpening, however Wavelet sharpening forces feature size ranges (octaves) upon the user, whereas the Sharp module lets the user freely choose a continuum of feature sizes that the user wishes to work on.

The Sharp module may also function as a scale decomposition tool, being able to selectively remove or attenuate features of a particular size. This allows multi-scale processing whereby different versions of the image with different scales may be processed independently and recomposed at a later stage.

Additionally, selective sharpening is also supported by means of a Mask. This way the unwanted sharpening of any features can be easily avoided.

Usage

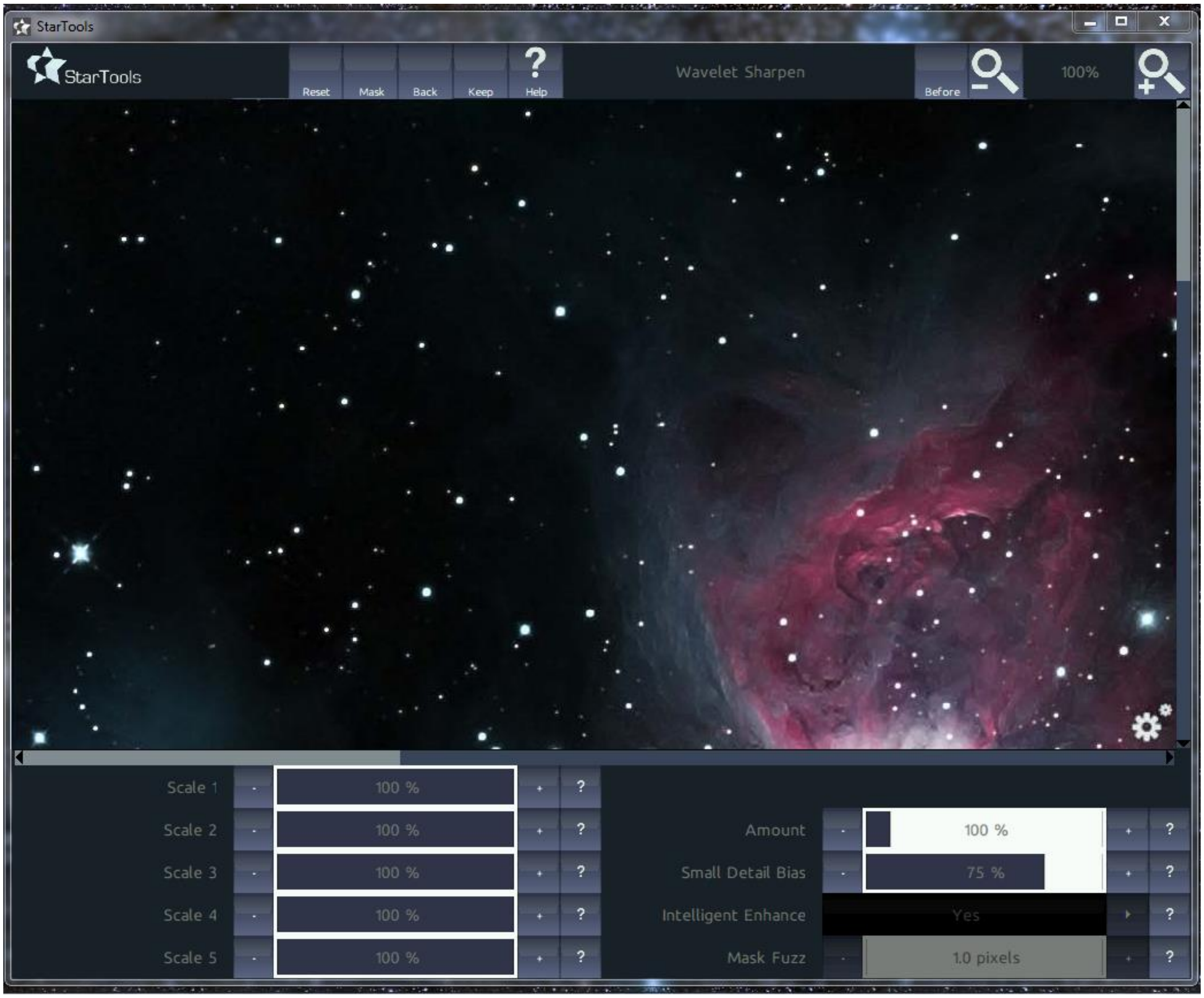


Illustration 20 – The Sharp module interface

This screen lets you setup the parameters that WAVELET SHARPEN should use to decompose the image into multi-scale components. Click the ‘Next’ button to commence decomposition. Upon completion you will be able to perform real-time wavelet sharpening.

Screen 1

- [STRUCTURE SIZE] specifies what size range the WAVELET SHARPEN module should cover.

Screen 2

- [SCALE 1] controls the amount of fine feature sharpening. This is the scale that governs features such as single pixel detail.
- [SCALE 2] controls the amount of medium to small feature sharpening.
- [SCALE 3] controls the amount of medium feature sharpening
- [SCALE 4] controls the amount of medium to large feature sharpening.
- [SCALE 5] controls the amount of large feature sharpening.
- [SMALL DETAIL BIAS] parameter specifies how the WAVELET SHAREN module should resolve cross-scale sharpening conflicts. It may be that sharpening in a larger scale causes contrast reduction on a smaller scale – a problem inherent to wavelets. This parameter allows for assigning a bigger importance to small scale detail. The larger the value, the more important the WAVELET SHARPEN module will treat small scale detail.
- [AMOUNT] allows throttling the strength of the sharpening across all scales. This parameter acts as a multiplier for the individual scale parameters.
- [BRIGHTNESS MASK POWER] specifies how much (if at all) sharpening should be avoided in darker parts of the image.

  A [BRIGHTNESS MASK POWER] of 0.0 applies sharpening to all pixels, bright and dark, evenly.

  A [BRIGHTNESS MASK POWER] of 1.0 increases the effects of sharpening progressively as pixels get brighter, leaving the very darkest pixels alone.

  A [BRIGHTNESS MASK POWER] bigger than 1.0 progressively applies sharpening to only the brightest pixels.

  TECHNICAL EXPLANATION: this parameter functions exactly the same as the [BRIGHTNESS MASK POWER] parameter in the LAYER module. What the WAVELET SHARPEN module does is the same as putting the unprocessed image at the foreground, with the sharpened image as the background. The corresponding BRIGHTNESS MASK MODE is WHERE FG IS LIGHT, USE BG.

- [CHANNELS] specifies if sharpening should be applied to just the color information, the brightness information or both.
- [MASK FUZZ] specifies the kernel radius of an optional Gaussian blur, to be applied to the mask. If a mask is used to selectively sharpen the image, using this parameter will ensure smooth (and hence undetectable) transitions between sharpened and unsharpened parts of the image. This option is unavailable if no mask is active (e.g. when all pixels in the whole image are tagged for processing).

Deconvolution

Overview

The DECONVOLUTION module reverses the effects that atmospheric turbulence has on your data. StarTools DECONVOLUTION implementation is unique in several ways;

Firstly, StarTools 'tracking' feature allows you to apply deconvolution on data you have already stretched. You can now use deconvolution at any stage during your processing.

Secondly, the regularization algorithm (the algorithm that keeps noise from propagating) is designed so that no guessing is needed as to how to set its parameters correctly – it is always set to provide the optimal balance between noise and detail.

Thirdly, the de-ringing algorithm in StarTools deconvolution implementation is not only very effective, it can actually resolve the singularities that overexposed star cores constitute and treat them like points of light.

To avail of de-ringing, simply create a mask with the objects that are causing ringing masked out (i.e. non-green).

To speed up rendering, you may specify a temporary image crop to work on by selecting a region with the mouse. To select the whole image for processing again, click the 'All' button.

There are two modes for the deconvolution algorithm in StarTools. There is one mode for Deep Space images with stars and another for lunar and planetary images. The latter expands the dynamic range as needed to accommodate highlights in the new deconvoluted image.

Usage

- (RADIUS] specifies the size of the blur that DECONVOLUTION will attempt to remove.
- [REGULARIZATION] specifies the balance between detail, noise and smoothness. A value of 1.0 specifies an optimum balance, a value less than 1.0 specifies more detail at the expense of more noise, whereas a value less than 1.0 specifies a smoother image at the expense of detail.
- [ITERATIONS] specifies how many iterations the DECONVOLUTION algorithm will through. Iterations beyond a certain point will not yield a better result, so only increase this value if further improvement can be seen. Please note that increasing this parameter will make rendering take longer.
- [IMAGE TYPE] selects between a special mode for Depp Space and Lunar / Planetary images. The latter mode frees up dynamic range for any deconvoluted highlights. These highlights may require extra dynamic range in the image than was available in the non-deconvoluted image. The Lunar / Planetary mode allocates this dynamic range as needed so that detail in the highlights is preserved.
- [MASK BEHAVIOR] specifies how a mask is used to enhance your image.

  The default value of ‘De-ring Mask Gaps, Hide Result’ still deconvolutes masked-out parts but applies de-ringing to them, so that the ringing in these parts doesn’t ‘bleed’ into other (masked-in) parts. The result of the de-ringing is then hidden by copying the non-deconvoluted image where mask gaps exist. In this case the MASK FUZZ parameter gradually lets the de-ringed result poke through.

  The ‘De-ring Mask Gaps, Show Result’ de-rings the parts of the mask that were not set, but does not hide the result. The parts that are de-ringed are shown wholly.

  Normal has the usual effect of retaining the original pixels where no mask was set. No de-ringing is applied in this instance.
- [MASK FUZZ] specifies the kernel radius of an optional Gaussian blur, to be applied to the mask. If a mask is used to selectively sharpen image, using this parameter will ensure

smooth (and hence undetectable) transitions between sharpened and unsharpened parts of the image. Refer to the MASK BEHAVIOR parameter’s help for an explanation of how the MASK FUZZ parameter interacts with the de-ringing algorithm.

Wipe

Overview

The WIPE module is one of the more powerful modules in StarTools.

Its main purpose is to eliminate unwanted light in an image. This unwanted light may come in the form of gradients, colour casts, light pollution, vignetting, amp glow or even a combination of all five

- Gradients are usually prevalent as gradual increases (or decreases) of background light levels from one corner of the image to another.
- Colour casts are a tint of a particular colour which, contrary to a gradient, affects the whole image evenly.
- Light pollution is the presence of a persistent haze of (often) coloured light, caused by urban street lighting.
- Vignetting manifests itself as the gradual darkening of the image towards the corners and may be caused by a number of things.
- Amp glow is caused by circuitry heating up in close proximity to the CCD, causing localised heightened thermal noise (typically at the edges). On some DSLRs and Compact Digital Cameras, amp glow often manifests itself as a patch of purple fog near the edge of the image.

Illustration 21 – M101, courtesy of Charles Kuehne

The image in Illustration 21 suffers from three sources of unwanted light-a gradient starting at the upper right corner, light pollution in the form of the typical yellowish light emitted by typical urban sodium lamps, and vignetting, clearly seen in the darkening of the corners.

Illustration 22 – M101, processed by StarTools Wipe

Wipe works by calculating a model of the unwanted light and then subtracting it from the image. The result is an image that is free from the unwanted light sources -as can be seen in the image in Illustration 22, Wipe has eliminated the unwanted sources of light while keeping the actual subject intact.

Usage

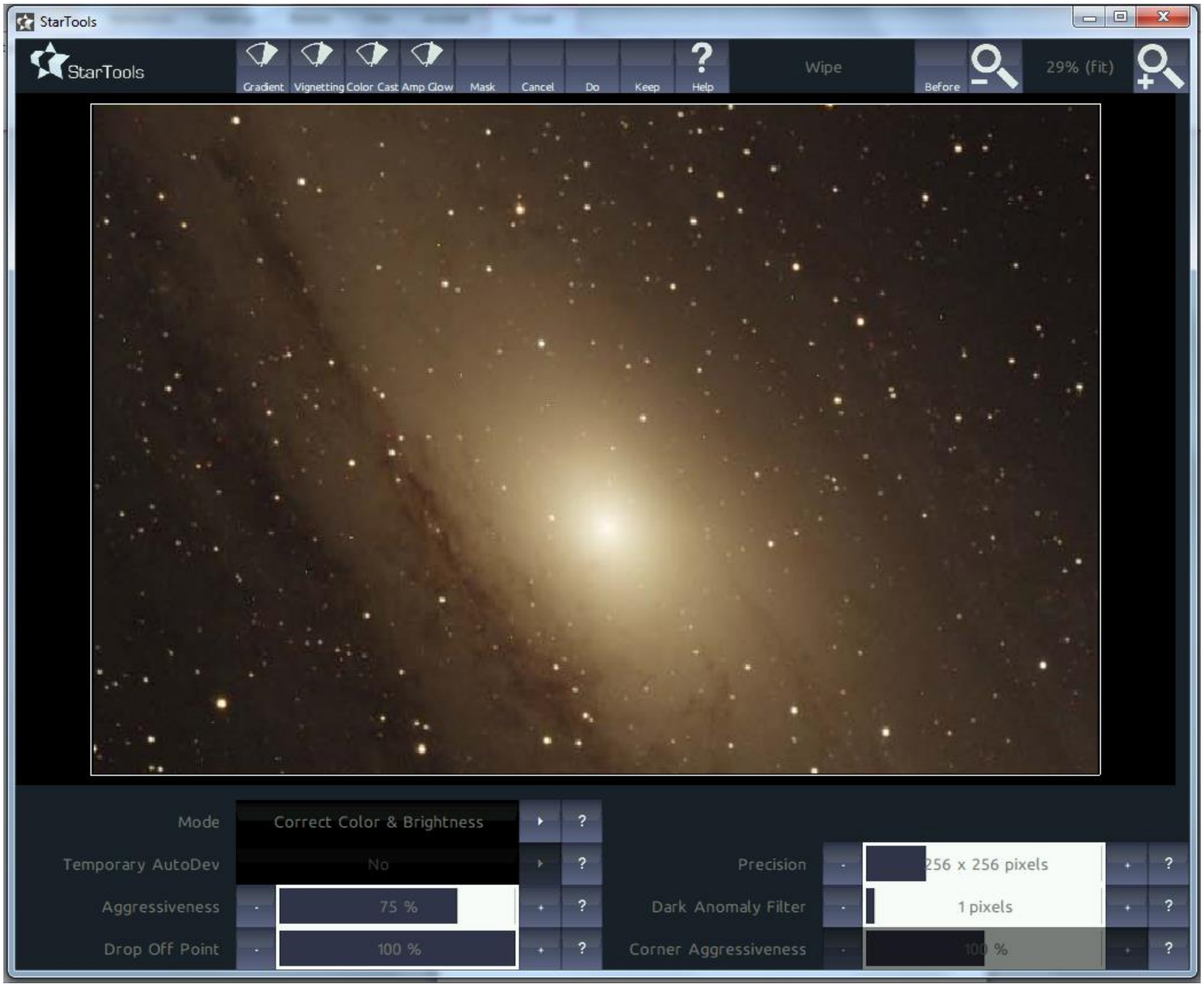


Illustration 23 – The Wipe interface

NOTE: WIPE usually does a great job unaided, however anomalies that are darker ('dark anomalies') then the galactic background (such as dead pixels, stacking artifacts, dust particles on the CCD, or terrestrial objects such as trees, mountains, etc) may throw WIPE's gradient detection off in some cases. In case of the presence of these dark anomalies, please remove them if possible, or create a mask with the pixels of these dark anomalies set to off (i.e. non-green) – make sure none of their pixels are selected for processing.

NOTE: When a mask is used, WIPE will still wipe the complete image, but will no longer take the masked-out pixels into account while building the gradient model.

TIP: If any dark anomalies are small (e.g. dead pixels), you may use the [DARK ANOMALY FILTER] parameter.

TIP: Be sure to inspect the wiped image for halos. If you see any, chances are they are caused by a 'dark anomaly'. Maybe you missed a dead pixel or something else that is 'darker-than-the-galactic-background'.

- [MODE] specifies which type of gradients are to be removed by WIPE.
  - [CORRECT COLOR & BRIGHTNESS] removes both color and uneven lighting gradients. This setting is best used for light pollution removal.
  - [CORRECT BRIGHTNESS ONLY] removes uncolored gradients; it only darkens the image in places where gradients are prevalent, but does not affect the color.
  - [CORRECT COLOR ONLY] neutralizes color gradients but does not affect the brightness levels (luminance) of an image.
- [OUTPUT GRADIENT ONLY] specifies if WIPE should output only the gradient calculated.
- [AGGRESSIVENESS] specifies how aggressive WIPE should be when removing gradients. The higher this value, the more aggressive WIPE becomes when removing local gradients.
- [DROP OFF POINT] specifies at which point from the center (i.e. how close to the corners) the CORNER AGGRESSIVENESS parameter comes into effect. Using a setting of less than 100% will let WIPE apply a different aggressiveness (CORNER AGGRESSIVENESS) to the corner of the image, which allows for the correction of vignetting.
- [CORNER AGGRESSIVENESS] specifies how aggressive WIPE should be when removing gradients in the corners of the image. The higher this value, the more aggressive wipe becomes when removing local gradients in the corners. Use this parameter to increase gradient removal in corners, for example to remove vignetting.

  NOTE: this parameter only takes effect when DROP OF POINT is reduced from 100%. It is grayed out otherwise.

- [TOP END TREATMENT] specifies what WIPE should do with the very brightest pixels when subtracting the gradient model.
    - [BRIGHTNESS MASK] use a brightness mask to keep the very brightest pixels in the processed result the same as in the original image. The upside is that no clipping occurs, but the downside is that faint remnants of any color cast may remain in the brightest pixels.
    - [WIPE 2.0] truncates all channels to the smallest value found for the new maximums of red, green and blue. The upside is that all colors will be correct, but the downside is that in extreme cases color clipping may occur (harsh color transitions).
- [PRECISION] specifies the amount of samples WIPE takes from the image. In large images with quickly changing gradients, higher precision may be appropriate.
- [DARK ANOMALY FILTER] specifies how strong WIPE should filter the image in order to get rid of any 'dark anomalies'. WIPE usually does a great job unaided, however anomalies that are darker ('dark anomalies') than the galactic background (such as dead pixels, stacking artifacts, dust particles on the CCD, or terrestrial objects such as trees, mountains, etc.) may throw WIPE's gradient detection off in some cases. The DARK ANOMALY FILTER can remove small dark anomalies such as dead pixels, small dust particles or small scratches.
- [DARK ANOMALY HEADROOM] specifies what to do with dark anomalies found by the DARK ANOMALY FILTER. This parameter specifies how much headroom is reallocated for them. More specifically it specifies how many gray levels they may take up as a percentage of the total amount of ray levels available.

  NOTE: this parameter only takes effect when DARK ANOMALY FILTER is not 'off'. It is grayed out otherwise.

[CAP GREEN] removes any dominant green color information from the image, relying on the fact that not many objects in outer space are green. The WIPE module can therefore be instructed to assume that any dominant green information is the result of noise or unwanted gradients.

- [TO YELLOW] caps green to yellow. If you find that this introduces yellow tints into your image, try TO BROWN.

- [TO BROWN] caps green to brown.

  NOTE: Exceptions of green areas are those with dominant O III emissions. Examples may include the Orion Nebula's core (M42) and the Tarantula Nebula (NGC2070).

Wipe comes with 4 handy preset buttons that quickly call up settings that are appropriate for common situations. These buttons are 'Gradient' for the removal of gradients, 'Vignetting' for the removal of gradients and vignetting, 'Color Cast' for the removal of general color casts and ‘Amp Glow’ which removes amp glow and requires a mask.

While Wipe performs quite well on most images with minimal user intervention, sometimes it needs direction on which parts of the image contain unwanted light and which parts of the image do not constitute unwanted light, such as, for example, galaxies, nebulas, or in the case where trees and other terrestrial objects make up part of the scenery.

For those cases it may be appropriate to create a Mask in the mask editor which specifies which pixels of the image Wipe should sample for unwanted light and which pixels of the image Wipe should ignore. Green pixels (e.g. the pixels that are 'on' in the mask) will be sampled in order to reconstruct the background levels and any gradients. Pixels that are not green in the mask (e.g. pixels that are 'off') will not be sampled by Wipe and will not influence the modeled gradients.

When masking out objects, please make sure that every pixel of that object is masked out (i.e. not 'green'/on). When in doubt, click the 'Shrink' button in the Mask editor a few times to make sure only background pixels are tagged.

Please note that usage of a Mask in Wipe may impact processing times somewhat, depending on how many pixels are masked out (e.g. are not green in the mask). This is normal behaviour. To find out more about Masks and how to create them with ease, please refer to the Masks chapter - page 150.

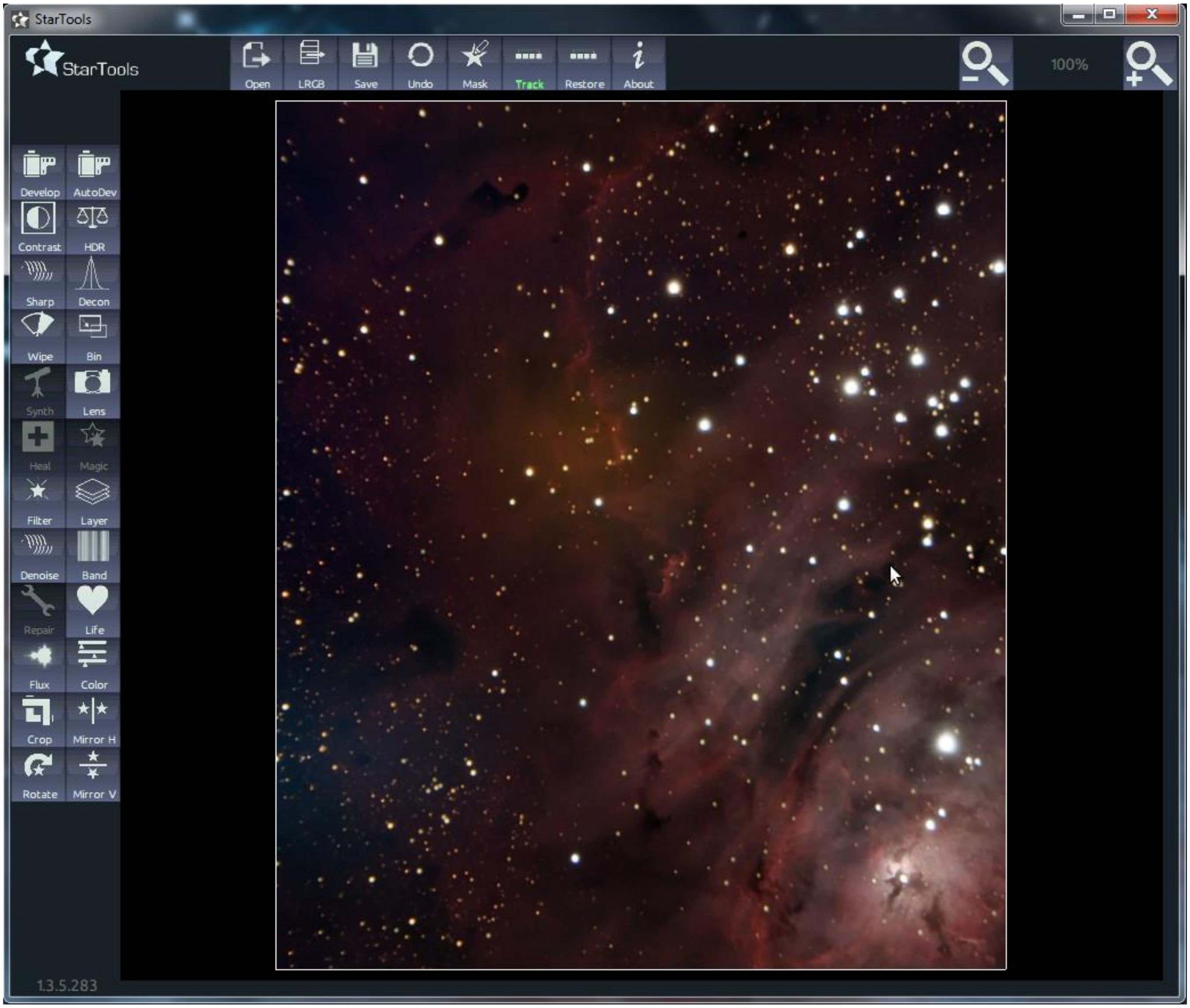


Illustration 24 – Two artifacts introduced by Wipe – a result of 2 patches of hot pixels

Bin

Overview

The BIN module offers the user a way to trade off resolution for a better signal-to-noise ratio (SNR). This algorithm is similar to the ‘low light mode’ and enhanced ISO modes found on the latest consumer compact cameras and DSLRs, with the exception that the StarTools BIN algorithm still yields correct results at the arbitrary sizes (not just powers of 2) by applying an anti-aliasing filter at the proper cutoff frequency corresponding to the new image size.Far from being a simple scaler, the Bin module offers arbitrary sized fractional binning of pixel data.Typical image scalers do not reduce noise. BIN has 4 preset buttons for 25%, 35%, 50% and 71% reductions.

Similar software algorithms are found in the more recent consumer digital cameras, often labeled 'low light mode', allowing for very high ISO values to be achieved at the expense of resolution. The StarTools Bin algorithm however still yields correct results at arbitrary sizes by applying a Nyquist filter at the proper cut-off frequency, whereas 'low light mode' typically forces a fixed ratio of 4:1 (also known as 2x2 binning) and in some cases 16:1 (also known as 4x4 binning).

When should you consider software binning

You should consider software binning when your image's scale is bigger than is needed to reveal the smallest detail. A typical example is an image that is recorded at a higher resolution than atmospheric seeing conditions permit. In this case the image will seem blurry. Reducing image size will reduce blur, retain detail and reduce noise. A common mistake is to retain too high a resolution during processing, when the scale of the smallest detail simply does not warrant the extra resolution[9]the image is said to be oversampled. The redundant pixels may instrumental in reducing noise and improving the signal's fidelity at a smaller scale that matches the scale of the smallest achievable detail.

In a sense, binning is the ultimate noise reduction tool. Whereas any other noise reduction tools necessarily alter the image based on educated guesswork, binning uses actual recorded (but otherwise superfluous) data to reduce noise.

[9] A notable exception is when the imager intends to apply a deconvolution algorithm, in which case the oversampled data is used to recover finer detail than is otherwise available in the original oversampled data.

Making the decision to perform binning very early on in the processing (before stretching levels and/or digital development), will improve the signal you get to work with and prevent noise from propagating.

A notable exception is when the imager intends to apply a deconvolution algorithm, in which case the oversampled data is used to recover finer detail than is otherwise available in the original oversampled data.

Finally, you could also consider software binning as a last resort to reduce noise, even if detail is removed. In this case, (with reduced data) binning may make the difference between a usable and an unusable data set by augmenting data at the expense of resolution.

Please note that some dedicated CCDs offer binning in hardware (typically only a 2x2 mode). Hardware binning will yield an even better signal-to-noise ratio, as the hardware binning also reduces read-out noise, which software binning does not. Depending on your subject and site conditions, hardware binning may make the difference between a usable image and a noisy pixel soup. In these conditions where read-noise mitigation is of the utmost importance, hardware binning will yield significantly better results than post-acquisition software binning.

Depending on your CCD characteristics, hardware binning may also have some drawbacks, such as diminished well depth or complete loss of colour in case of a One Shot Colour (OSC) device. Of course, software binning is an optional post-processing step, whereas hardware binning is performed during capture and is irreversible.

Usage

- The Bin module takes a single parameter 'Scale', whose value can be read in several ways as it relates to several aspects of the new image versus the old image. SCALE defines the reduction in resolution and the corresponding desired increase in signal to noise ratio. Also displayed is the noise reduction factor and the bit-depth improvement that corresponds to the selected SCALE factor.

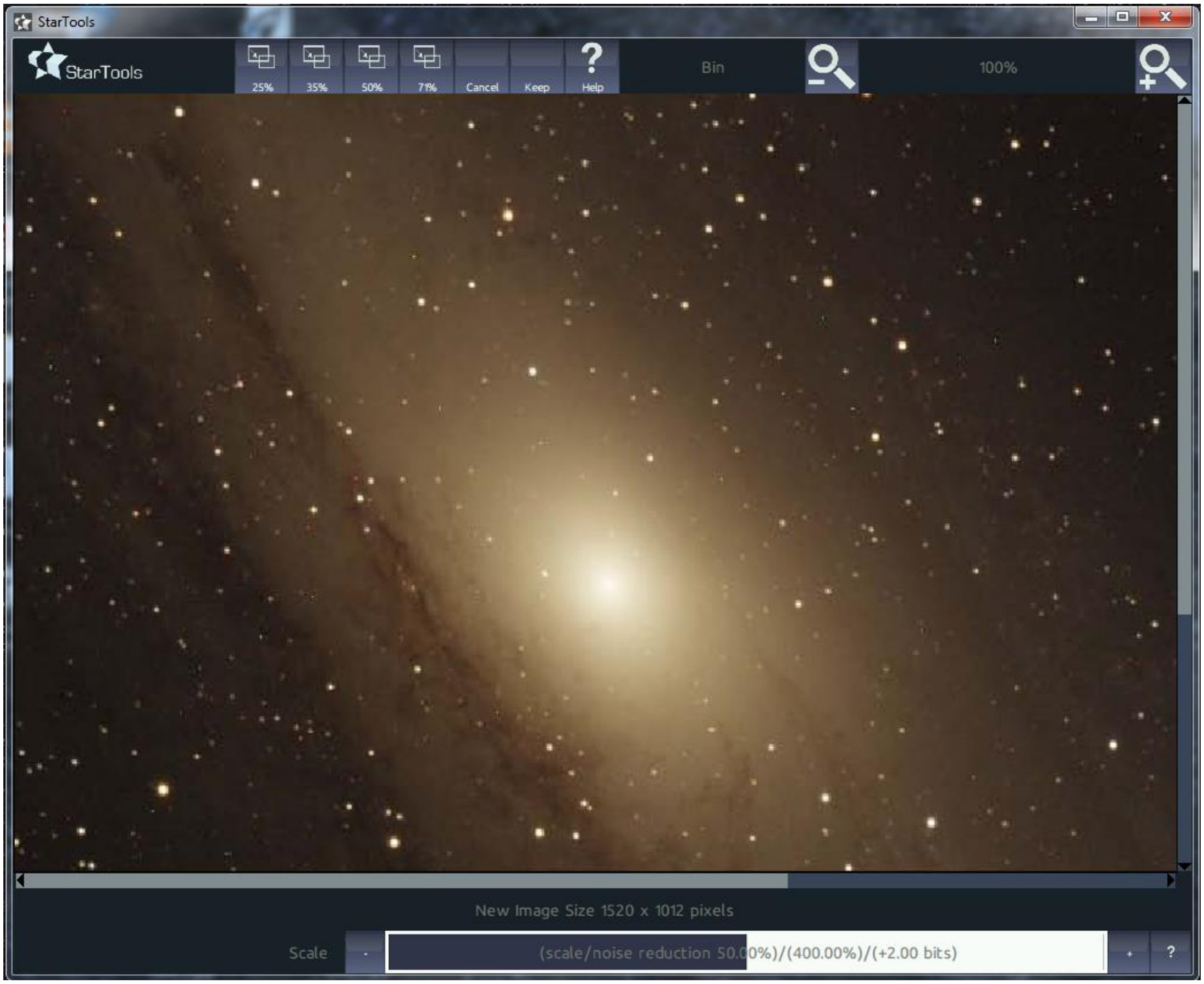


Illustration 25 – The BIN module interface

As can be seen in Illustration 25, the value indicator in the level setter control reads "(scale/noise reduction 50.00%)/(400%)/(+2.00 bits)". This may be read in the following ways

The image's X and Y dimensions have been reduced to 50% of their respective original values.

Noise has been reduced to 50% of its original level.

The original image was 400% the size of the newly binned image.

There has been an improvement of 400% to noise levels.

Bit-depth has been increased by 2.00 bits.

Notice how noise reduction does not increase linearly with image size reduction, i.e. to get more noise reduction, the image will have to be scaled down more and more.

Synth

Overview

The Synth module is one of the most complex (but gratifying) modules in StarTools. Synth allows the user to augment or even completely replace starlight by modeling new starlight, based on photometry information extracted from the image. Essentially, Synth reverses the full image train and models a new (virtual) one step-by-step.

The Synth module has several uses;

- Making stars brighter and tighter.
- Making stars look like they were imaged from orbit.
- Breathing life into 'flat' images by re-modeling physically correct starlight energy distribution.
- Modeling physically correct diffraction spikes.
- Modeling starlight as it would look through much bigger (and more expensive) telescopes or wholly different designs.

Synth performs best on images where it can perform accurate photometry – that is images that are of sufficient resolution, sharpness and with minimal noise. Please note that Synth does not modify anything else but stars and starlight.

Prerequisites

Before Synth can perform star synthesis, the user will need to provide it with two crucial items;

- A star mask, outlining where in the original image stars are located that need to be resynthesised.
- A virtual telescope model and its resulting energy distribution signature (also known as a Point Spread Function, or PSF).

If you do not provide Synth with a star mask and a PSF, Synth will display a message asking you to create them - Synth does not work without them.

Star Mask

The creation of a star mask can be accomplished in seconds using the 'Auto' function in the Mask editor (see the Mask chapter, page 150 - and how to quickly create a star mask - page 151). The Mask editor may be accessed through the 'Mask' button in the Synth module or from the main screen.

Please note that, while you're free to use Synth selectively on stars, the most natural looking results are achieved when all stars are treated equally by Synth, hence it is strongly recommended to simply use the Auto function in the Mask editor and select all stars.

While inspecting the Mask, make sure that only stars are selected, and not other bright objects, which Synth erroneously may treat like stars - deselect unwanted objects in the Mask editor.

Telescope modeling

Besides a star mask, a virtual scope model, is created in the scope modeler, as a second prerequisite.

Clicking the 'PSF' button in the Synth module will launch the scope modeler. The default settings will result in a model that should look similar to the one in Illustration 26.

To the left we can see a top down view of our virtual telescope. You will notice it is very similar to what you would see if you were to look straight down a Newtonian Optical Tube Assembly (OTA).

The right image represents the photon distribution of a point light (e.g. a star) as it is diffracted by the different components in the OTA. Note that this image is not necessarily a representation of a star, rather it serves as an indication of what individual stars will look like and how different OTA configurations influence the diffraction pattern as you design your virtual scope.

The virtual telescope model is controlled by 15 parameters;

- Aperture - specifies the diameter of the OTA. All other measurements and sizes controlled by the other parameters are relative to this parameter. Together with the

'Focal Length' parameter and the 'Image Diameter' in the Synth module, aperture significantly influences the concentration of starlight of individual stars in the final image. Focal Length -

specifies the length of the OTA. Together with the 'Aperture' parameter and the 'Image Diameter' in the Synth module, this parameter influences the concentration of starlight of individual stars in the final image. To calculate the focal ratio, divide this number by the aperture, e.g. for an aperture of 200mm and a focal length of 1200mm, this yields 1200/200 = f/6.

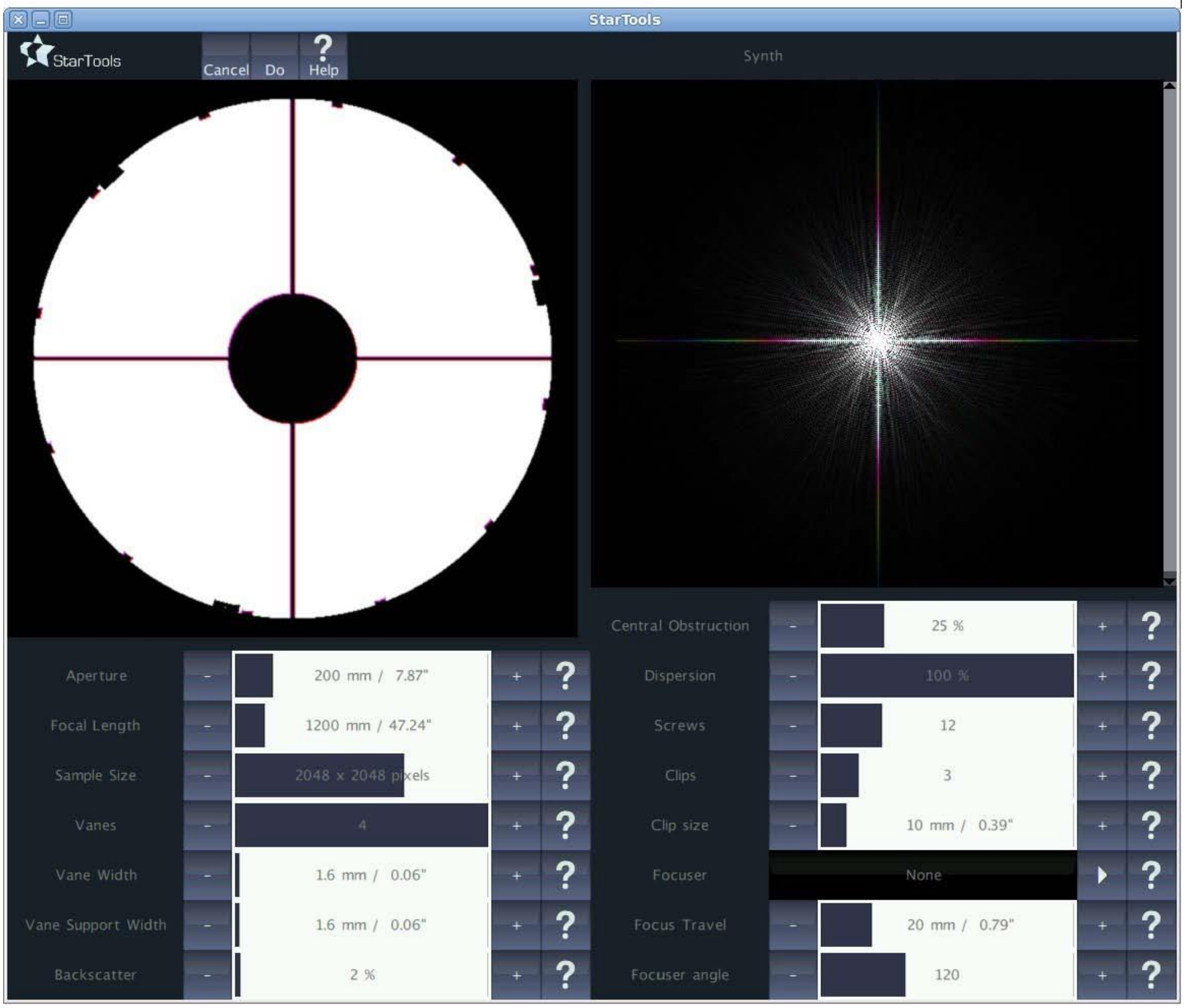


Illustration 26 - A virtual model of an 8" f/6 Newtonian design telescope (left) and its corresponding Point Spread Function (right).

- Sample Size - controls the fidelity of the model. Larger values will approximate energy distribution better and for larger areas, but will take longer to compute and render. In reality a star's starlight is scattered throughout the image (and even beyond), however these quantities of light are typically very minimal beyond a certain radius (the exception being the light of very long diffraction spikes). To speed up rendering, therefore we can usually get away with calculating the light scattering over a much smaller area ('Sample Size') around the star where it is actually noticeable. In the case of very bright stars, the area that the scattering is calculated for as specified by the 'Sample Size' parameter may not be sufficient to accommodate most of the starlight. If the latter is the case a clear boundary artefact will become visible where Synth stopped calculating scattered light. This boundary will often manifest itself faint squares around brighter star.

- Vanes - specifies the number of spider vanes that hold the 'Central Obstruction' (for example the secondary mirror on a Newtonian) in place. This parameter is responsible for the appearance of distinct diffraction spikes. The physical modelling dictates that

  1 vane will cause 2 fainter diffraction spikes.

  2 equally spaced vanes will cause 2 brighter diffraction spikes.

  3 equally spaced vanes will cause 6 fainter diffraction spikes.

  4 equally spaced vanes will cause 4 brighter diffraction spikes.

- Vane Width - specifies the width of the individual spider vanes. Most noticeably, this parameter influences the frequency with which the 'rainbow pattern' in any diffraction spikes repeats itself.

- Vane Support Width - specifies the width of any spider vane support mounted to the OTA, as found in some scopes. Most noticeably, this setting has some subtle effects on the central flare in brighter stars.

- Backscatter - specifies how much starlight is reflected by the primary mirror on to the various components in the OTA. This light is subsequently emitted as diffused light, having the effect of slightly reducing contrast and dulling down the brightness of any diffraction spikes and parts of any central flares.

- Central Obstruction - specifies the size of a circular central obstruction (typically a mirror holder) as a percentage of the full aperture. This settling greatly influences the diffraction pattern. Note that a setting of 0% (i.e. no obstruction) effectively turns the virtual telescope into a refractor design.
- Dispersion - specifies the strength of the dispersion effect - the subtle way in which different wavelengths (colours) of light are diffracted differently. Most noticeably, this setting controls the strength of the 'rainbow effect' in the diffraction spikes. Note that the physical modelling dictates that stars that do not emit all wavelengths equally, will generate 'rainbows' with different colours attenuated.
- Screws - specifies the number of screws that protrude from the OTA into the light path.
- Clips -specifies any mounting clips that keep the primary mirror or lens in place.
- Clip size -specifies the size of any mounting clips that keep the primary mirror or lens in place.
- Focuser -specifies the type of focuser that the OTA is equipped with. The presence of a focuser subtly affects the central flare's diffraction pattern, adding a faint but distinct spike. The different options are

  'None' - no focuser protrudes from the OTA into the light path.

  '1.25 inch' -a standard 1.25” focuser extends into the OTA.

  '2 inch' -a standard 2” focuser extends into the OTA.

  '3 inch' -a standard 3” focuser extends into the OTA.

- 'Focus Travel' -specifies how much of the focuser protrudes into the OTA.
- 'Focuser Angle' - specifies the angle at which the focuser is mounted to the OTA.

Once you are satisfied with the model, click the 'Do' button to commence the high resolution Point Spread Function calculation. Upon completion of the calculation, you will be returned to the Synth module.

Star Synthesis

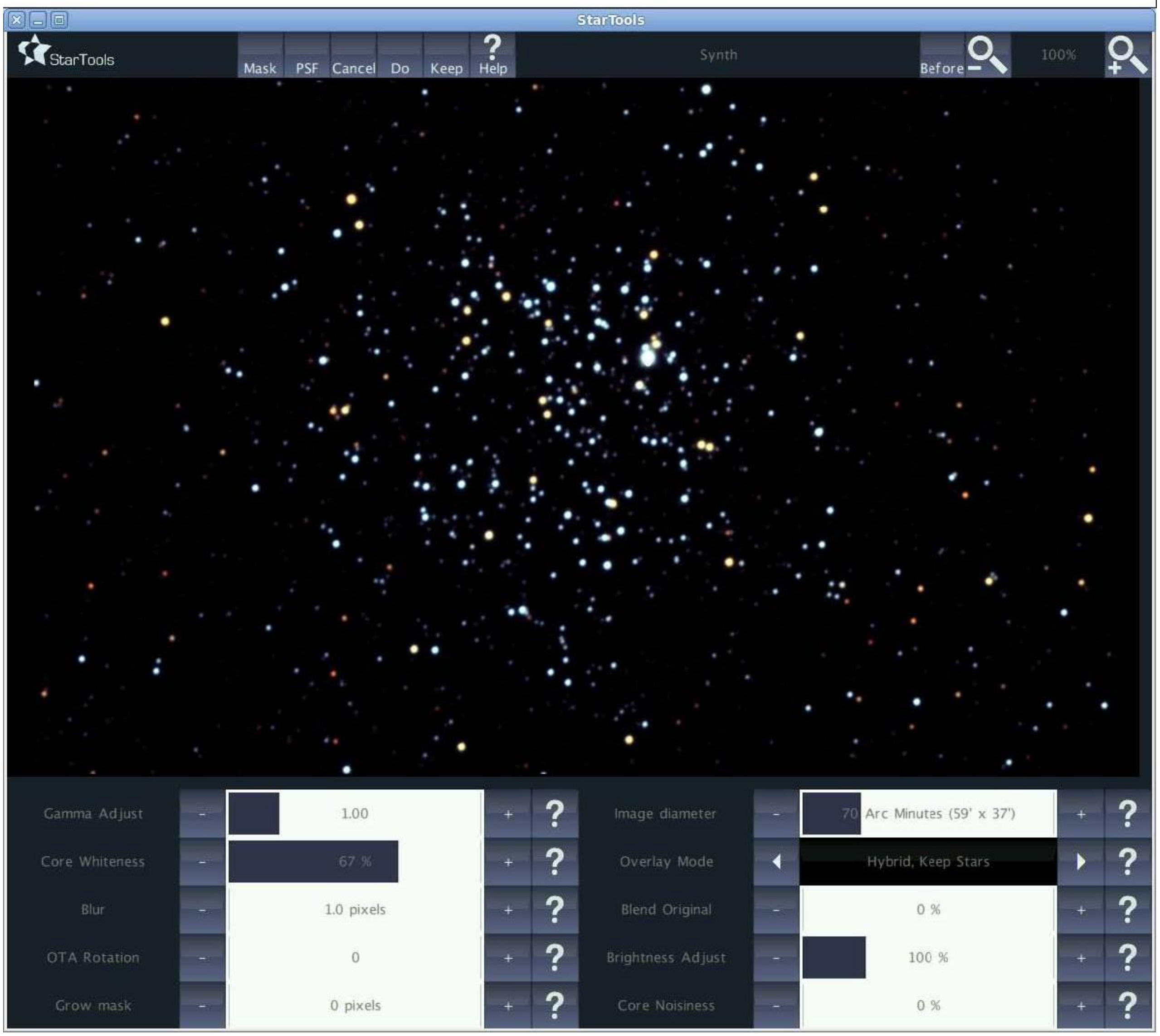


Illustration 27 - The Synth module interface with an image of M11 (image acquisition by Jim Misti).

Once a star mask and a scope model have been created, the parameters for star synthesis should be set up. Star synthesis itself may now be initiated by clicking the 'Do' button.

Star synthesis and rendering are controlled by 10 parameters. Please note that some of these parameters will apply in real-time, whereas some will only take effect once the 'Do' button is clicked. The following parameters are available;

- Gamma Adjust - specifies the gamma adjust that is applied to the synthesised star layer. This parameter is an excellent way to control the prevalence of any diffraction spikes and flares. Once a star layer is available (i.e. the 'Do' button has been clicked), this parameter updates in real time. Core Whiteness - specifies how much a star's color in the image influences its the color of its synthesised counterpart in the synthesised star layer. High values will tend to mostly white stars, whereas low values will tend to very colourful stars (however adherent to the colour spectrum of a glowing body). This parameter does not update in real time and is only applied upon the next rendering iteration triggered by clicking the 'Do' button.
- Blur - emulates blur due to atmospheric seeing conditions. It is highly recommended to try and match the appearance of the original image, in order to make the synthetic star layer blend in. A synthetic layer that is too sharp will likely stand out too much and be easily detectable as artificial. Once a star layer is available (i.e. the 'Do' button has been clicked), this parameter updates in real time.
- OTA Rotation - effectively specifies the rotation of the Point Spread Function relative to the image. Most noticeably, it controls the angle at which any diffraction spikes occur. This parameter does not update in real time and is only applied upon the next rendering iteration triggered by clicking the 'Do' button.
- Grow Mask - grows the active star mask by the specified amount of pixels. This parameter is particularly useful in 'Overlay Modes' that remove the stars for which new artificial ones are synthesised- increasing this value may help removing any traces of the old stars that are being replaced. This parameter does not update in real time and is only applied upon the next rendering iteration triggered by clicking the 'Do' button.

- Image Diameter - specifies the image dimensions in arc minutes. Synth uses this value in conjunction with the virtual scope model's aperture and focal length to determine the appropriate light concentration for the image and scope modelled.
- Overlay Mode -specifies how Synth should composite the new image from the original image and the synthesised star layer. Once a star layer is available (i.e. the 'Do' button has been clicked), this parameter updates in real time. The following modes are available;
- 'Hybrid, Remove Stars' - applies a proprietary compositing algorithm which is a hybrid between addition, lightening and screening pixel math. Any stars that are replaced are completely removed from the original, before layering of the new ones is performed.
- 'Lighten, Remove Stars' - applies 'Lighten Only' pixel math to composite the new image. Any stars that are replaced are completely removed from the original, before layering of the new ones is performed.
- 'Addition, Remove Stars' -adds the synthesised layer to the original, however any stars that are replaced are completely removed from the original, before layering of the new ones is performed.
- 'Hybrid, Keep Stars' - applies a proprietary compositing algorithm which is a hybrid between addition, lightening and screening pixel math. The original image (including any stars that are to be replaced) is kept and the synthesised layer is simply layered on top of it.
- 'Lighten, Keep Stars' - applies 'Lighten Only' pixel math to composite the new image. The original image (including any stars that are to be replaced) is kept and the synthesised layer is simply layered on top of it.
- 'Addition, Keep Stars' - adds the synthesised layer to the original. The original image (including any stars that are to be replaced) is kept and the synthesised layer is simply layered on top of it.
- 'Subtract, Keep Stars' - is an experimental mode where synthesised stars are subtracted from the original. This mode may help to bring out detail where fain detail is otherwise

drowned out by a star's glare. It is recommended to only use this mode with virtual refractor designs, combined with some blur.

- Blend Original - specifies what amount of the original image should make up the final image. Once a star layer is available (i.e. the 'Do' button has been clicked), this parameter updates in real time.

- Brightness Adjust -specifies a corrective factor to the amount of light all stars output. Once a star layer is available (i.e. the 'Do' button has been clicked), this parameter updates in real time.

- Core Noisiness - specifies a percentage which is used to push the image beyond unity (full white), effectively overexposing the image somewhat. This may aid Synth's star core detector to recognise noisy cores better – Synth assumes big stars' cores are white. When they are not perfectly white (for example due to noise or sharpening artefacts), Synth's core detector may fail. This parameter may help to overcome this issue. This parameter does not update in real time and is only applied upon the next rendering iteration triggered by clicking the 'Do' button.

After clicking the 'Do' button and subsequent synthesis has been completed, the screen image will update to reflect the newly synthesised layer, similar to the screen in Illustration 28.

Further tweaking and modification is now possible. Again, please be aware that some parameters only come into effect once the 'Do' button is clicked and synthesis has been initiated.

If you notice any artefacts in the form of distinct squares of light around brighter stars, then this is caused by a too low PSF sample size. There are multiple ways of fixing this; increasing the sample size, adjusting the gamma (lower) or increase the angular size value

If starlight seems to be over concentrated, adjust gamma, create a new virtual scope model (with a different aperture and/or focal length) or increase the angular size value.

Always scrutinise the image looking for errors - it is possible the Synth module did not correctly detect a star's size or did not correctly separate multiple overlapping stars, instead detecting them as a single large star.

Experimentation with the different blend modes may be in order for best results, deepening on the source image.

It is acceptable for local contrast to suffer somewhat, pushing up background levels. This is easily remedied by using the Contrast module.

Subsequent sharpening of the synthetic stars may also benefit your image.

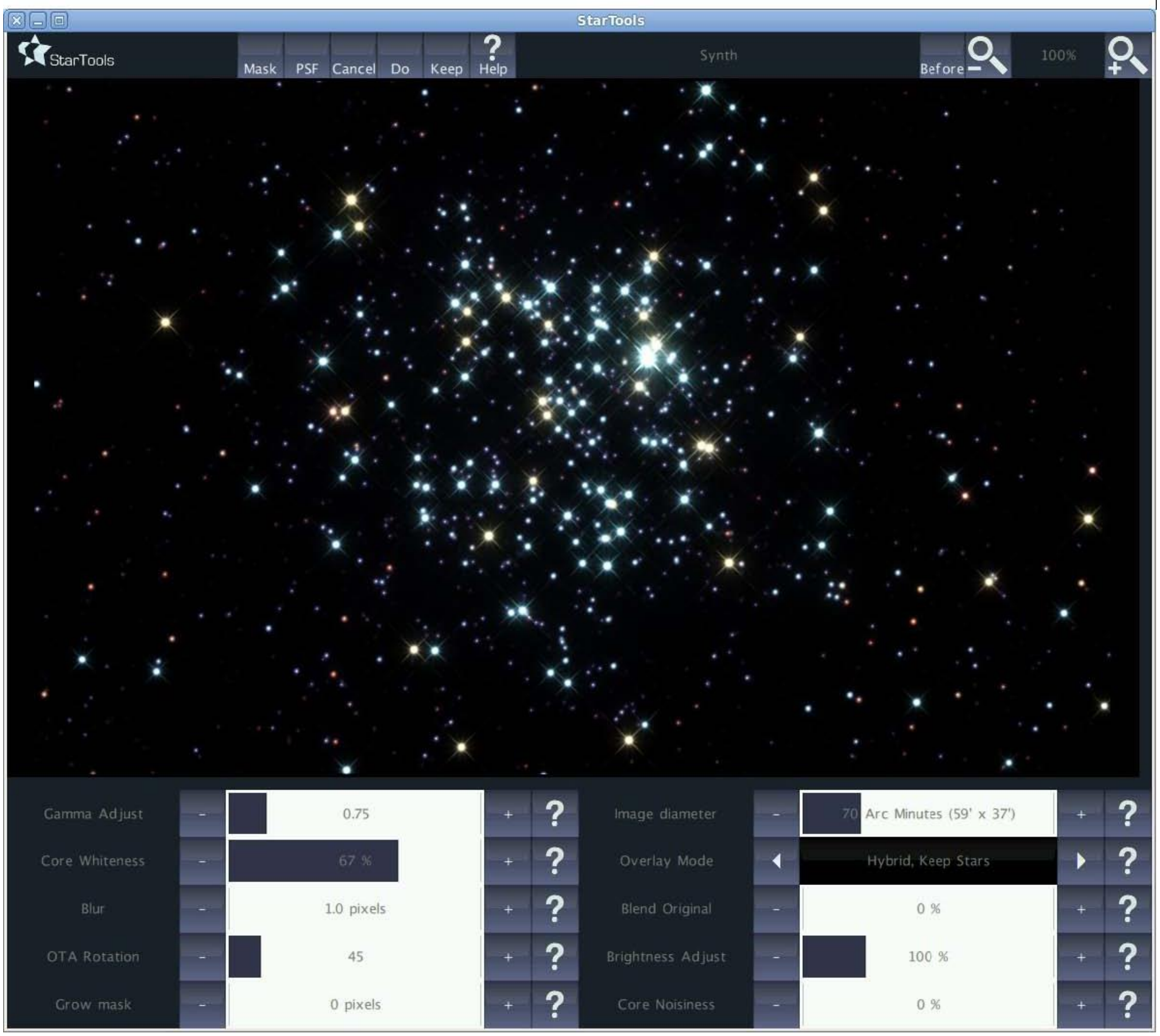


Illustration 28 - Synthesised image of M11, modelled with the 8” f/6 Newtonian PSF of Illustration 25.

Examining an output sample

Illustration 29 and Illustration 30 show a 3x magnified before and after crop of the image of open cluster M11, as seen in Illustration 27 and Illustration 28 respectively. We used the Point Spread Function (PSF), resulting from the virtual scope model of Illustration 26.

First off, it must be said that the original was of very good quality and did not stand much to gain from star resynthesis, other than perhaps the addition of the diffraction spikes if so desired. If the image however was of less quality, for example suffering from bad focusing, tracking issues or over-processing artefacts, etc., then the resynthesis would have made a bigger difference, in that the stars would now have been perfectly round, in focus and natural looking.

The virtual scope model of Illustration 26 describes an 8” f/6 Newtonian design telescope, having a secondary mirror holder, which is mounted in the OTA by four spider vanes. The four spider vanes have clearly caused 4 diffraction spikes (at a 45 degree angle), while the various other components have caused some subtle diffraction patterns and spokes emanating from the stars.

The concentration of stars in the middle has caused another diffraction effect- a local brightening of the area around it. This concentration of diffused starlight is regarded by some astrophotographers as desirable, as it imparts a certain depth to the image. This effect is most effective on globular clusters and wide-field shots, where star concentrations tend to correlate with nebulosity. While there are other techniques to approximate this effect, again this is a physically correct implementation and not an approximation.

We can observe that the different star colours are correctly reflected in the repetition of the 'rainbow' pattern; wavelengths (colours) that are less prevalent in the starlight are correctly attenuated.

Lastly, one thing that Synth did clearly enhance over the original, is the resynthesis and highlighting of the star cores of the smaller stars, which had been reduced to indiscriminate blobs without a clear core.

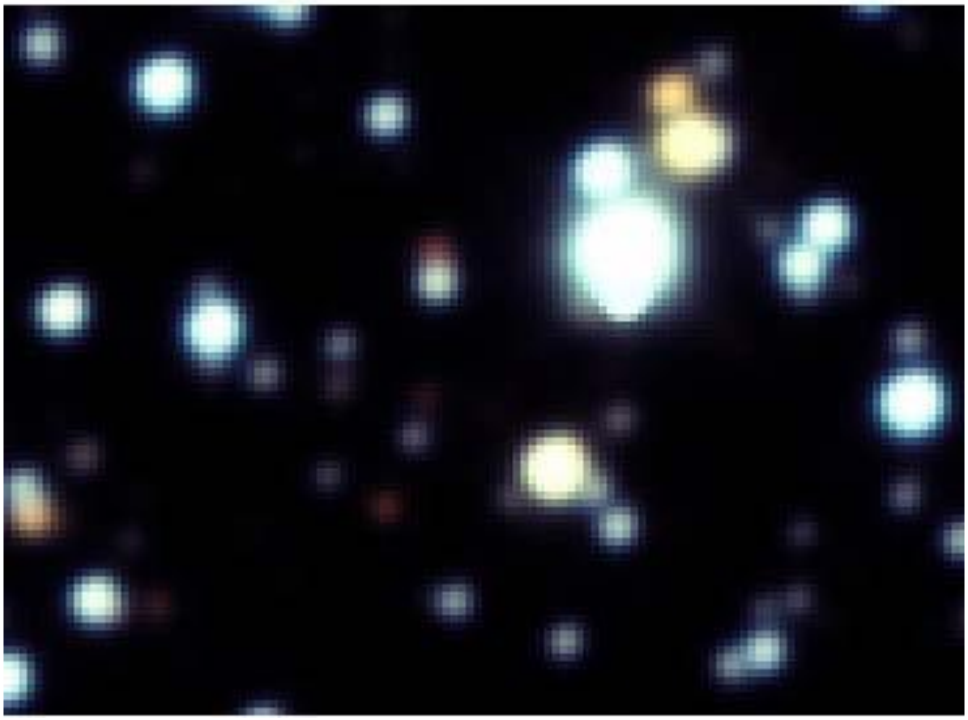


Illustration 29 - Crop of original image at 3x magnification.

Illustration 30 - Crop of synthesised image at 3x magnification.

The ethics of using synthesised components in your image

The desirability of deliberately introducing diffraction artefacts such as diffraction spikes and localised diffused starlight is up for debate and is down to personal preference. However, fact is that any diffraction artefacts that the user can introduce by Synth are physically correct and not an approximation.

Regardless, the Synth module reconstructs starlight based on photometry performed on the original image, measuring stars' magnitudes, color temperature and location. It is up to the user to specify constraints and to decide what (and how much) is removed and what (and how much) is added back in its stead.

Astrophotographer and author Jerry Lodriguss wrote an interesting article[10] on the ethics of astronomical image processing and image manipulation.

When you publish your image, it is common practice to mention the equipment, software and software plug-ins that were used to produce your image. You may find mentioning the use of the Synth module appropriate.

[10] http://www.astropix.com/HTML/J_DIGIT/ETHICS.HTM

Lens

Overview

The Lens module allows for correcting some of the common lens and mirror aberrations and defects, such as some types of chromatic aberration and coma. The curvature of the image resulting from the correction is automatically cropped. For best results, correct an image with LENS prior to cropping. This will make it easier to find the center of the distortion (often close to the middle of the image) and should make the procedure repeatable with the same parameters for any future images taken with the same setup under the same circumstances.

Chromatic aberration is the phenomenon of different wavelengths (colours) of light coming to focus at slightly different distances. Chromatic aberration is usually prevalent in cheaper lens systems, but even the most expensive optical systems suffer from a minute amount of chromatic aberration.

The Lens module can correct for some of the ill effects of lateral or transverse chromatic aberration. This type chromatic aberration often takes the form of unwanted colour fringes, progressively getting worse towards some (or all) corners of the image.

Please note that the Lens module does not correct for axial or longitudinal chromatic aberration. This type of chromatic aberration is often the cause of blue or purple halos around stars. To correct for this type of chromatic aberration, please refer to the Filter module chapter and example - page 99.

Coma is an inherent property of telescopes using parabolic mirrors. Light from a point source (such as a star) in the center of the field is perfectly focused at the focal point of the mirror. However, when the light source is off-center (off-axis), the different parts of the mirror do not reflect the light to the same point. This results in a point of light that is not in the center of the field looking wedge-shaped. The further off-axis, the worse this effect is. This causes stars to appear to have a cometary coma, hence the name.

The Lens module can also correct for coma by applying a reverse distortion to the image.

Usage

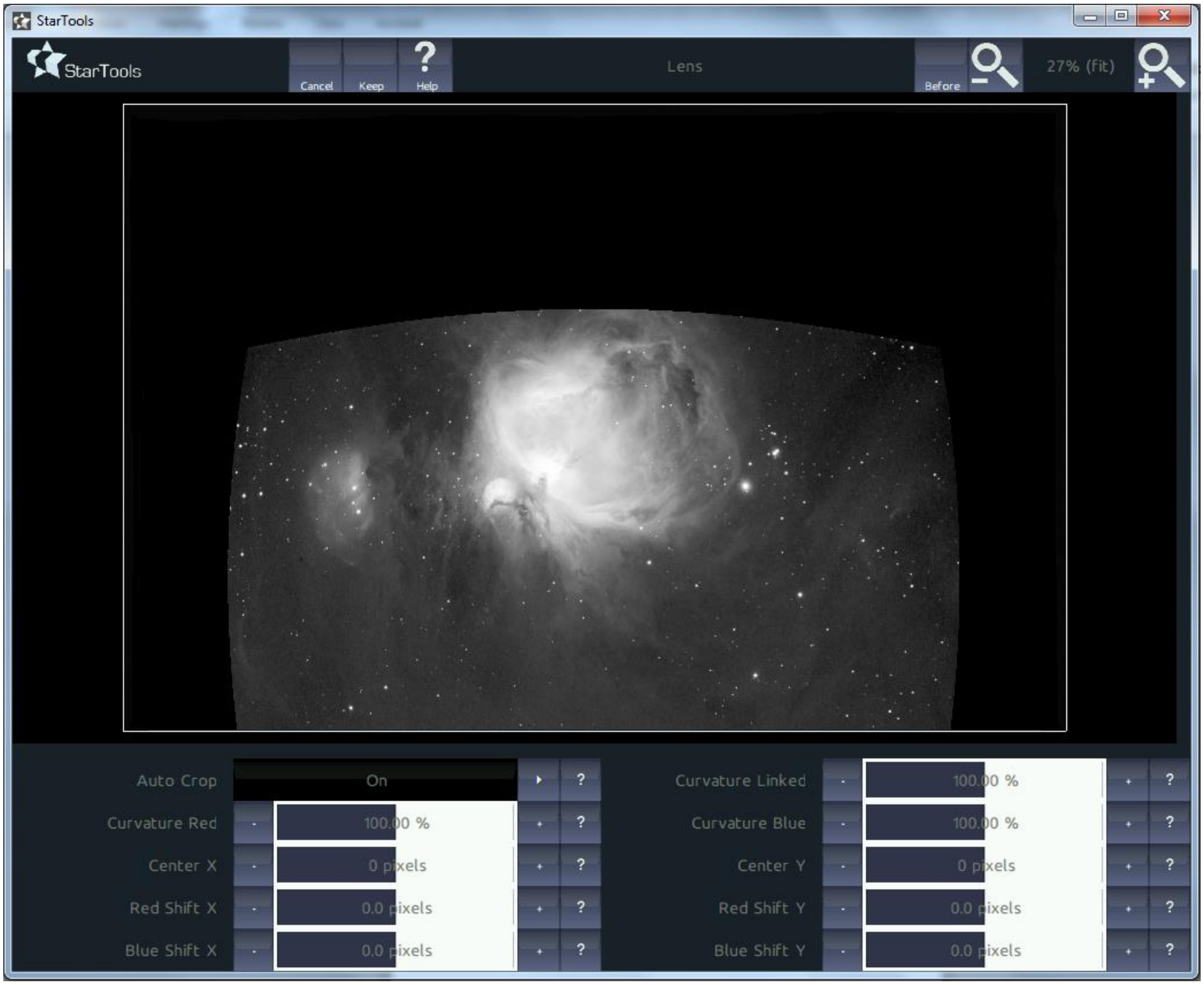


Illustration 31 – LENS module interface.

The Lens module has 10 adjustable parameters and settings;

- Curvature Red–adjusts the field curvature of the red channel. For coma correction/field flattening, simply keep both CURVATURE RED and CURVATURE BLUE equal or use the CURVATURE LINKED parameter.
- Curvature Blue -adjusts the field curvature of the blue channel. For coma correction/field flattening, simply keep both CURVATURE RED and CURVATURE BLUE equal or use the CURVATURE LINKED parameter

- Center X - specifies the central X-axis coordinate of the principal point (i.e. the center of the distortion).
- Center Y - specifies the central Y-axis coordinate of the principal point (i.e the center of the distortion).
- Red shift X -specifies the amount of (fractional) pixels to shift the red channel's pixels in the horizontal direction.
- Red shift Y -specifies the amount of (fractional) pixels to shift the red channel's pixels in the vertical direction.
- Blue shift X - specifies the amount of (fractional) pixels to shift the blue channel's pixels in the horizontal direction.
- Blue shift Y - specifies the amount of (fractional) pixels to shift the blue channel's pixels in the vertical direction.
- Auto crop - toggles the auto cropping feature on or off. Because a distorted image may curve/bulge when it is corrected, not all pixels in the final (square) image may be populated. Auto cropping will crop the image in such a way that all pixels are populated. This will mean, however, that some pixels at the edges may be lost as a result.
- Curvature Linked – adjusts the field curvature of both the red and blue channels simultaneously.

To correct coma (e.g. perform field flattening), simply adjust the Red Radius and Blue Radius by the same amount until stars no longer appear elongated.

Heal

Overview

The Heal module removes unwanted pixels from an image and substitutes new ones, based on neighbouring pixels. The Heal module is particularly useful for removing stars from an image, or defects such as scratches, dead pixels, hot pixels, etc. Please note that the Heal module is not particularly useful for removing 'noisy' pixels as tagged by the Auto feature.

Usage

The Heal module is simple in its operation and only takes 3 parameters;

Grow Mask - specifies an additional amount of pixels the currently active mask should grow by. The growing is applied temporarily and does not affect the active mask. This parameter is useful when the active mask does not completely cover all pixels that require removal, for example when you are trying to remove stars from an image and star halos remain.

- Algorithm - switches between two algorithms;
- 'Gradient' - fills any patches of pixels that are marked for removal with a smooth gradient. This algorithm is perfect for removing small amounts of pixels. However, since the substituted pixels are void of detail, this algorithm may not be appropriate for large areas.
- 'Inpainting' - attempts to fill any patches of pixels that are marked for removal with detail, based on neighbouring pixels.
- Second Pass -specifies whether heal should perform a second pass, after checking for pixels which were generated (healed) lighter than in the original. In the case a pixel was generated which was lighter than the original, the original pixel is used, instead of the generated pixel. This step helps generating better fits for missing pixels in the case where stars are being removed.

Using Heal to process stars independently

Let's say we have this image here;

Illustration 32 – Heart Nebula

Let's create a star mask that Heal can use to heal out the stars; (Mask, Auto, Stars, set 'Exclude Color' to 'Red' so that it won't select much of the nebula);

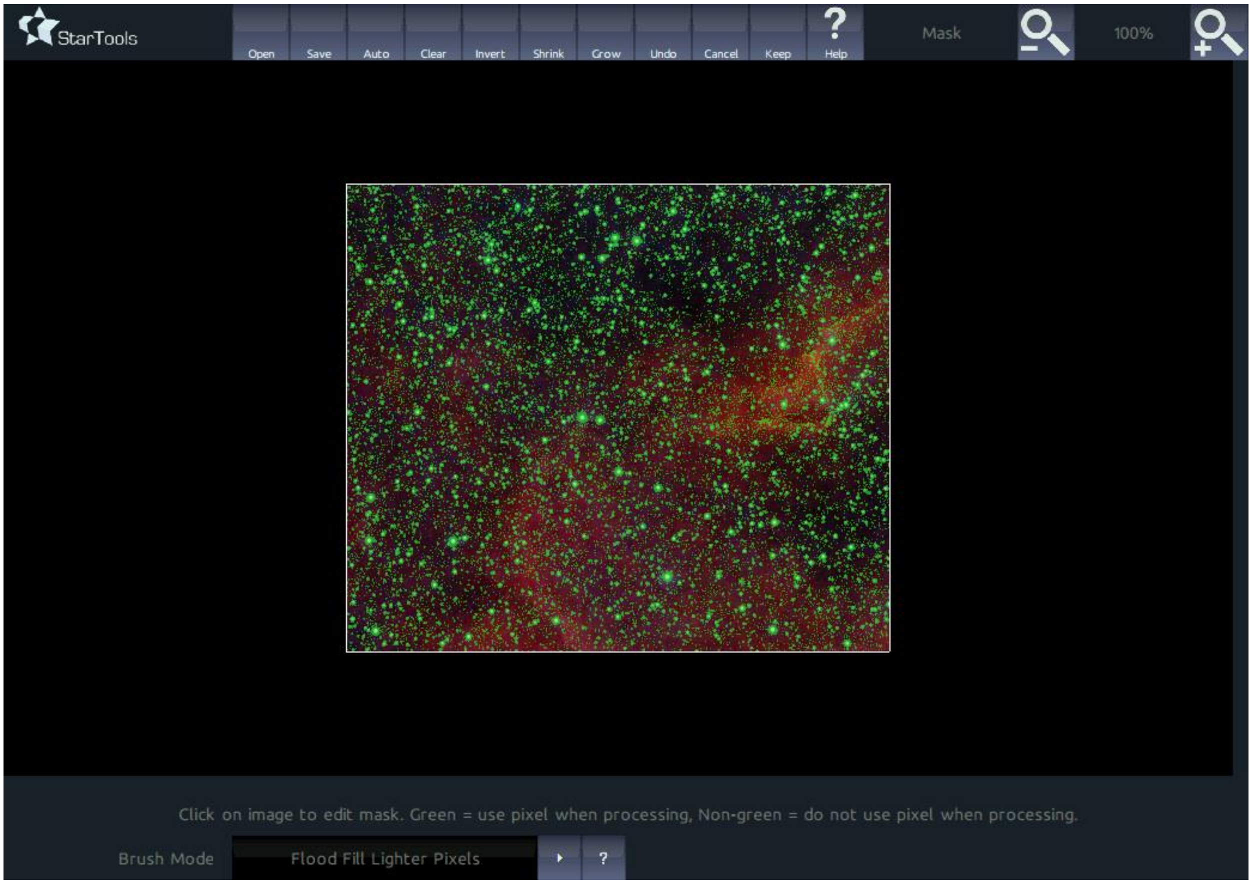


Illustration 33 – Creating a mask

Now run the Heal module, but make sure you set 'New darker than old' to 'Yes'. What this does is guarantee that replacement pixels for the stars are always darker than the stars themselves;

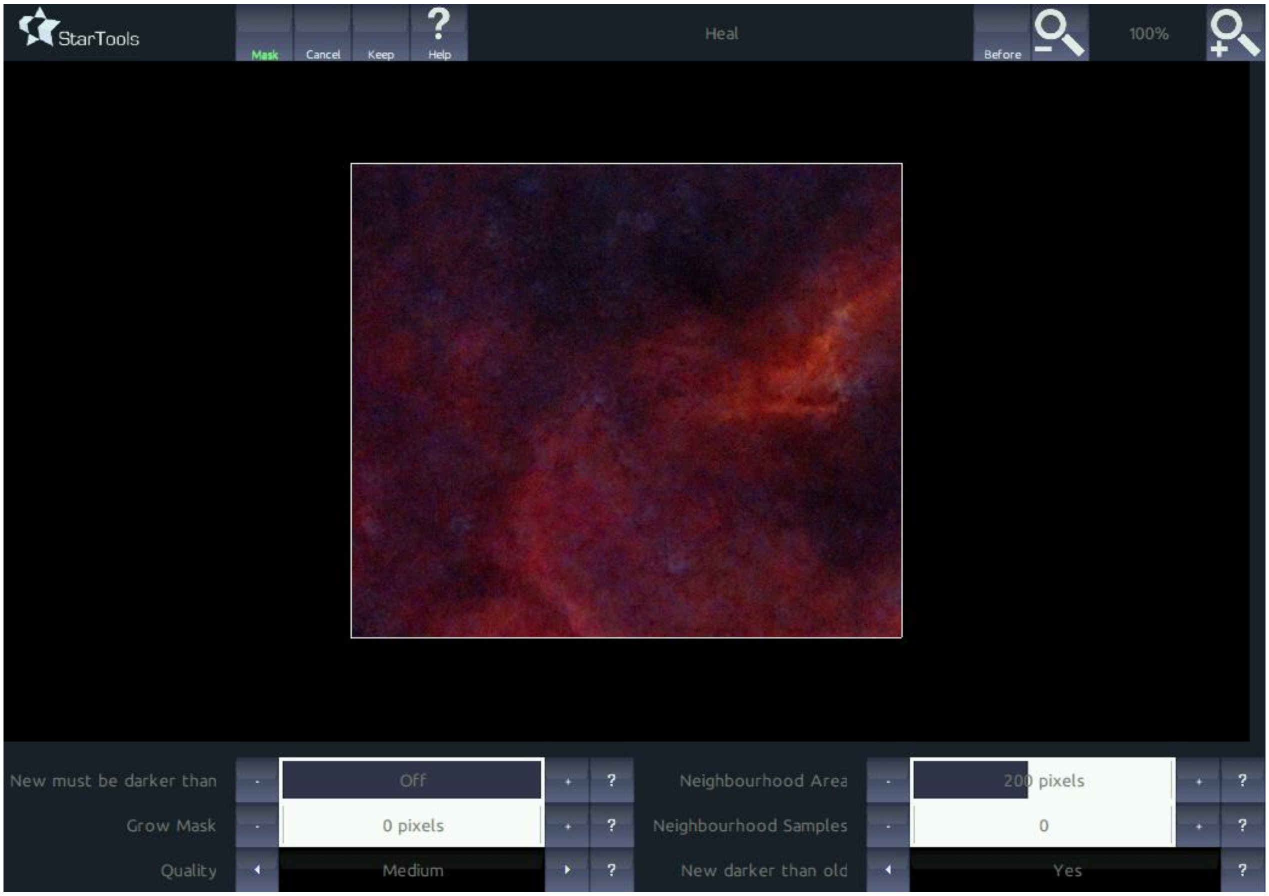


Illustration 34 – The resultant image

'Keep' the result and save the healed image for future reference. Now reset the mask (Clear, Invert) and launch the Layer module.

In the Layer module, click Undo->Bg. This will set the background to what's in the undo buffer (which is the original image).

Now set Layer Mode to 'Subtract' and voilà! We have perfectly extracted the stars;

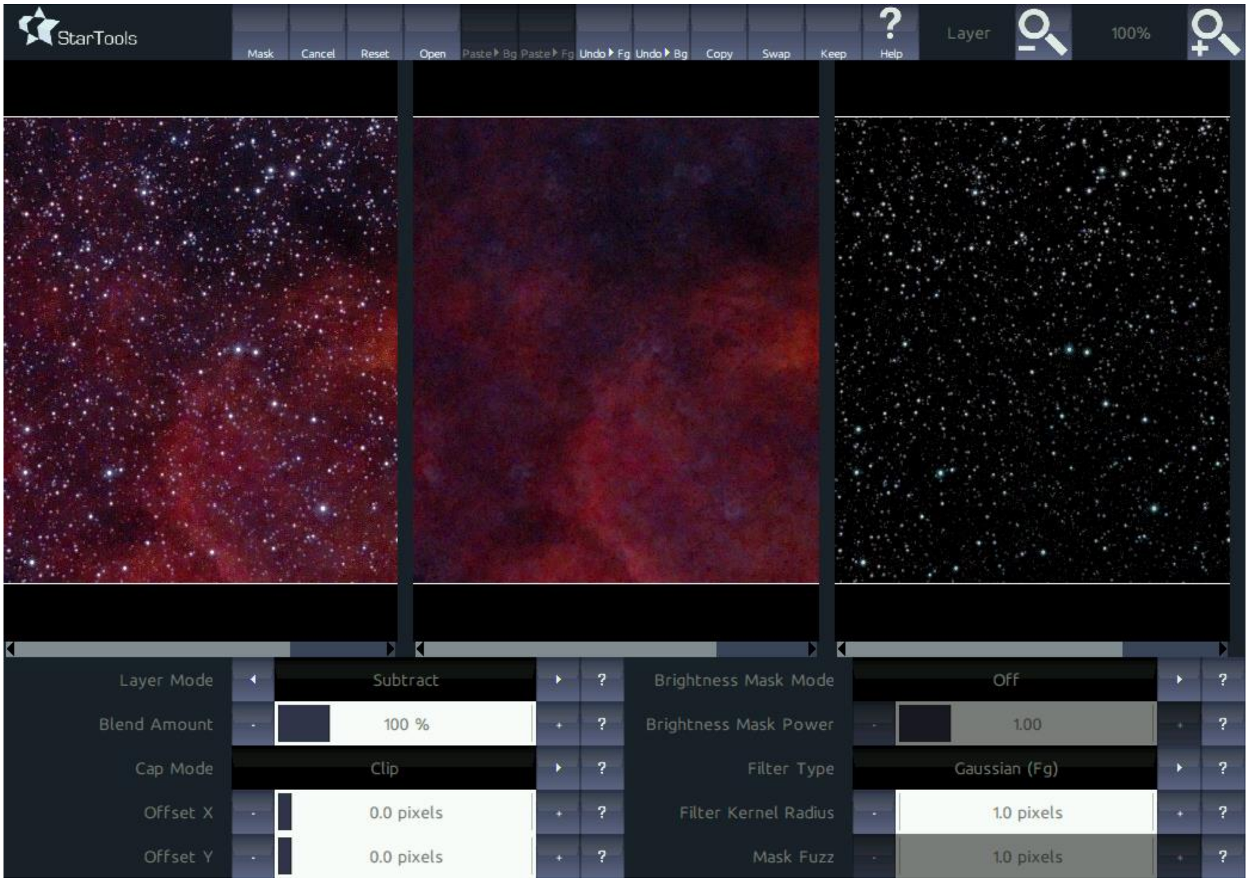


Illustration 35 – In the Layer module

Keep the result and save the star field - we'll need it later to layer it back in.

Click 'Undo' to restore the background-only image.

Now let's have some fun with the background!

For example, let's do some wavelet sharpening of the larger scale details;

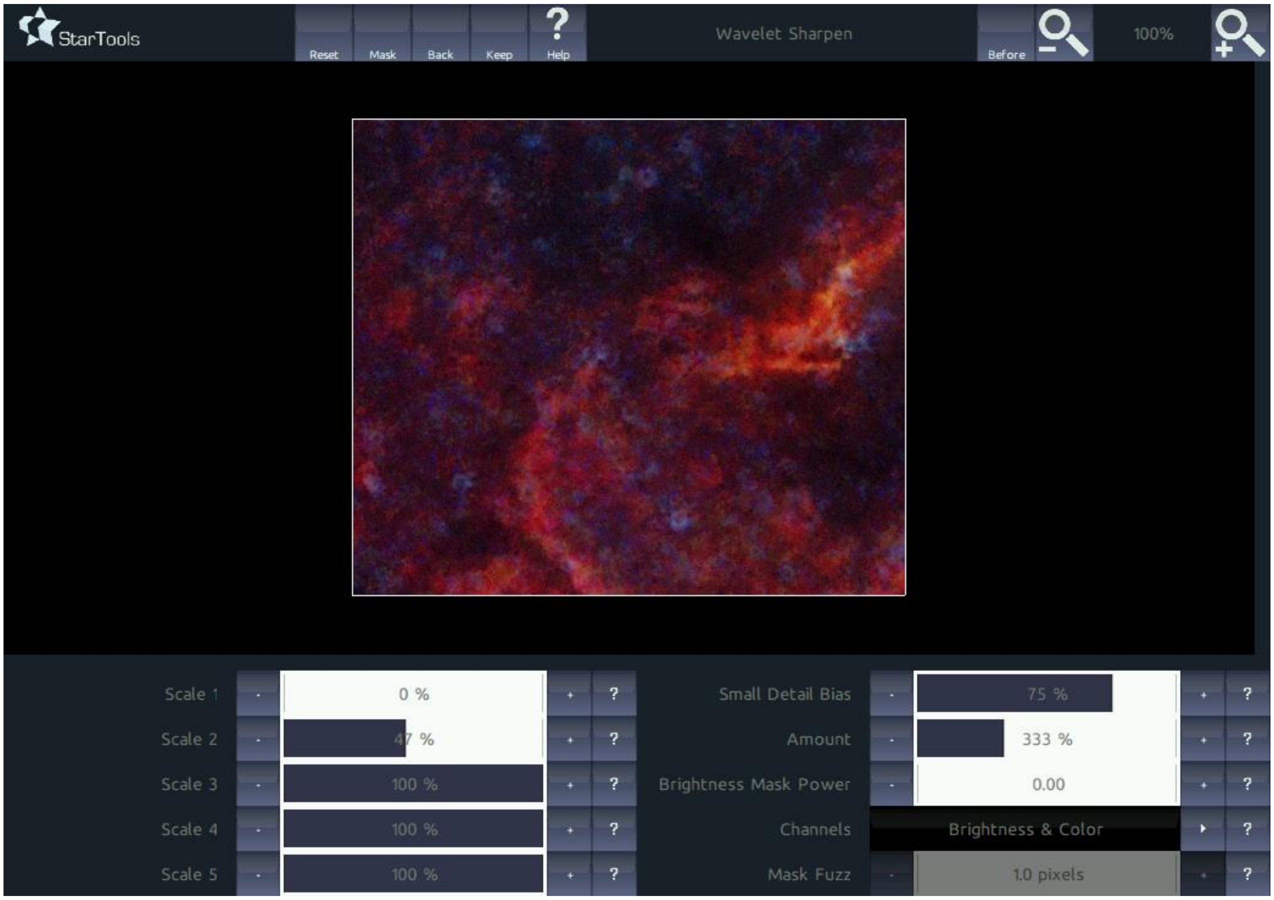


Illustration 36 - Sharpening

There's plenty more we could do (Life, Wavlet Denoise, HDR, etc.) but for the purpose of this demo this is enough.

Once we're happy with our new background, all we need to do now is layer the stars back in. Launch the Layer module, click Open, locate and load the stars-only image.

All we need to do now is simply 'Add' the stars back to the background. And Bob's our (proverbial) uncle!

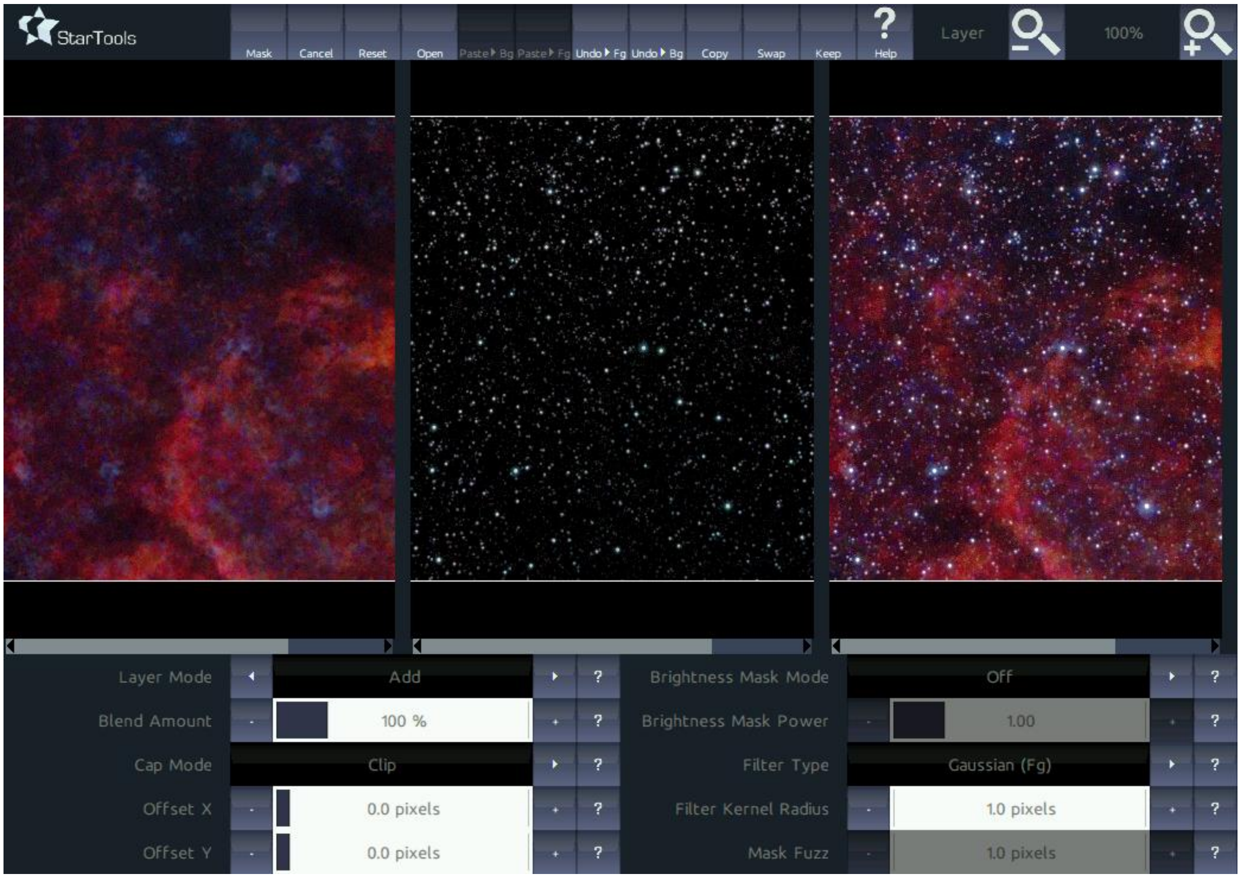


Illustration 37 – Adding the Stars back

Note that stars may be slightly blown out in places where nebula+starlight exceeds unity (= pure white). This should be fairly minimal, but if that is a concern to you, set 'Cap Mode' to 'Normalize'.

Illustration 38 – The completed image

Magic

Overview

The Magic module gives the user control over the appearance of stars in the image during post-processing. Various operators are available.

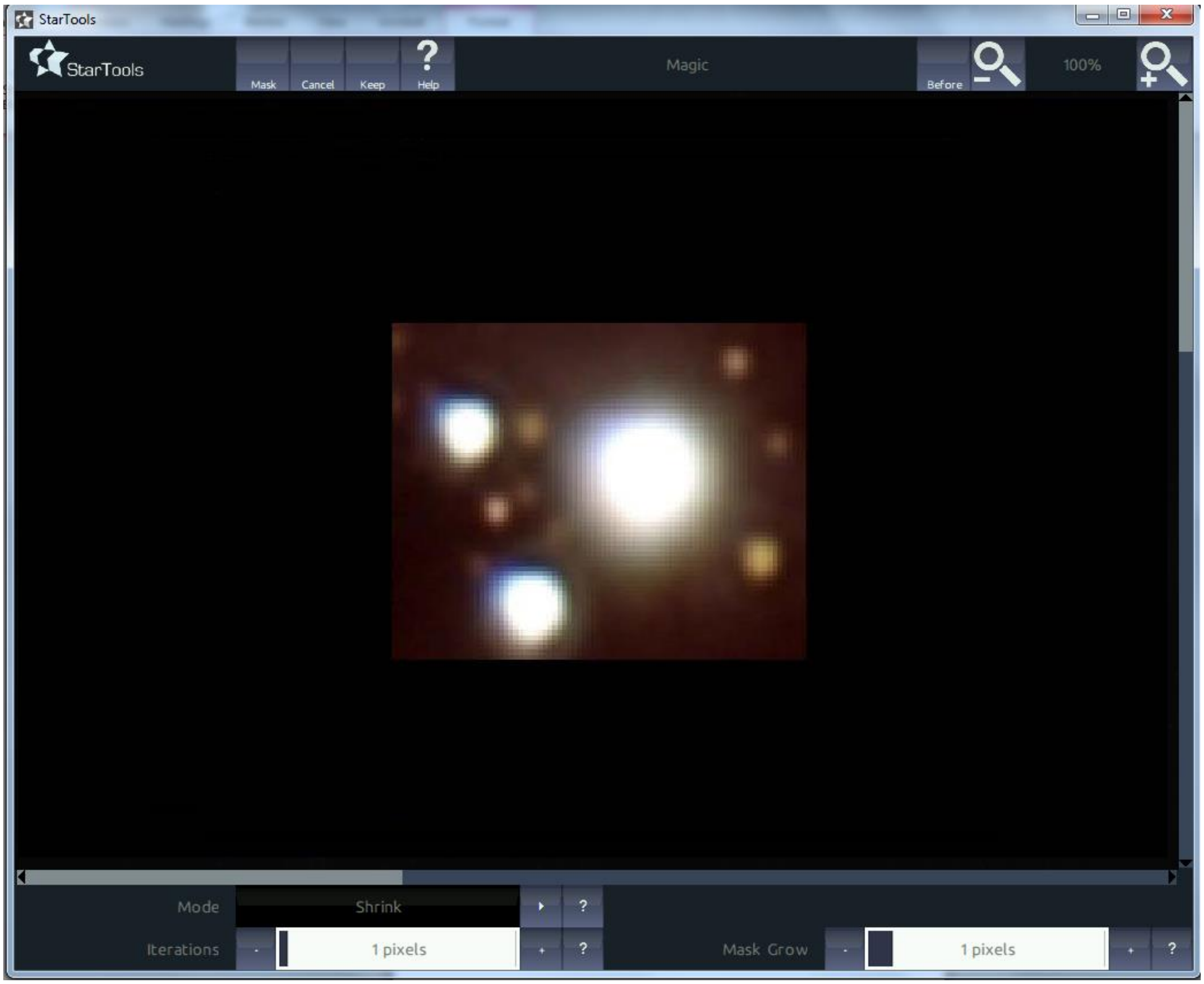


Illustration 39 - A test image with no further processing (all parameters set to 0). The big white star has been selected as our subject by means of a mask.

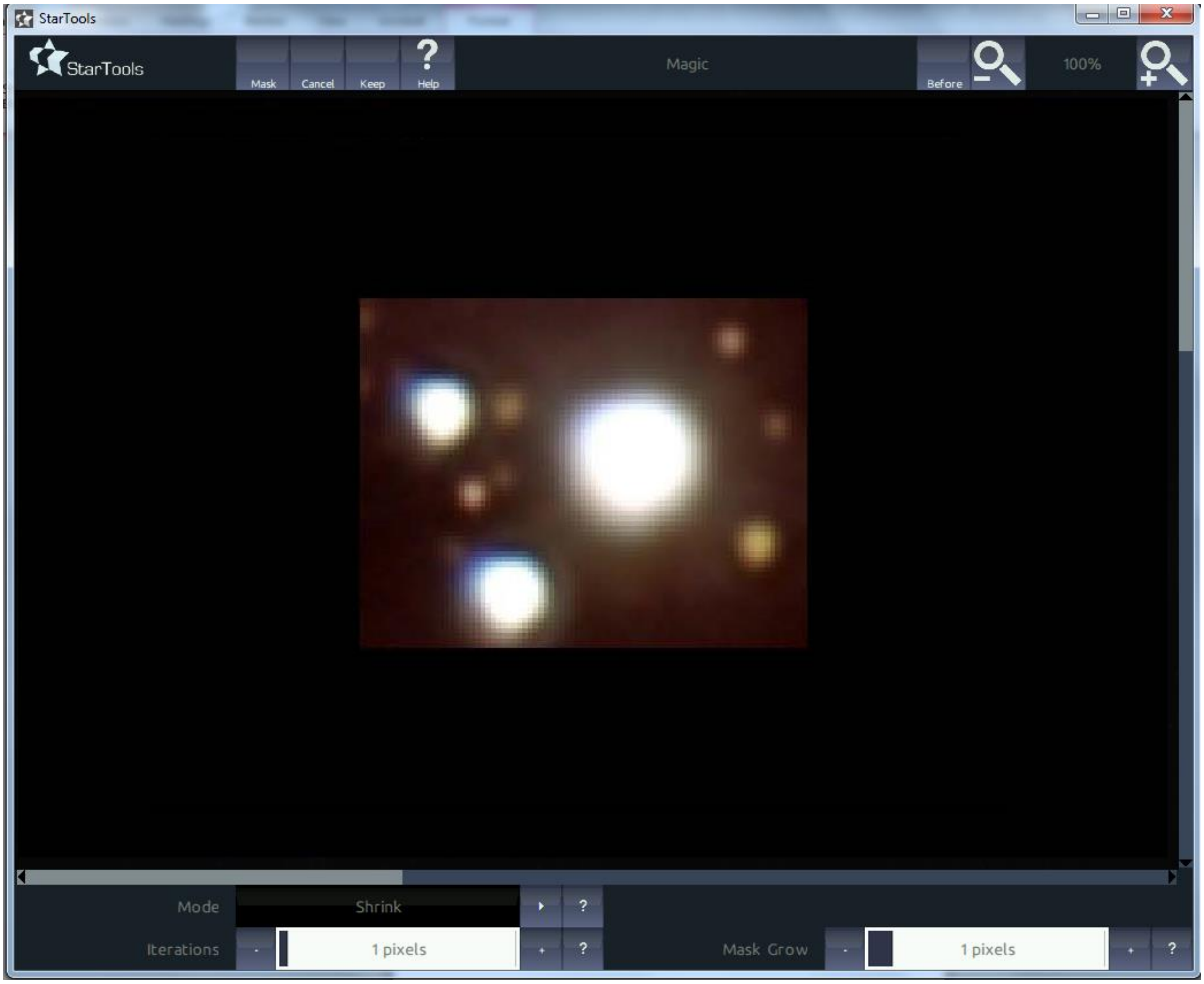


Illustration 40 - The 'Tighten/Debloom' operator in effect. Notice how the star's halo/glare has been diminished.

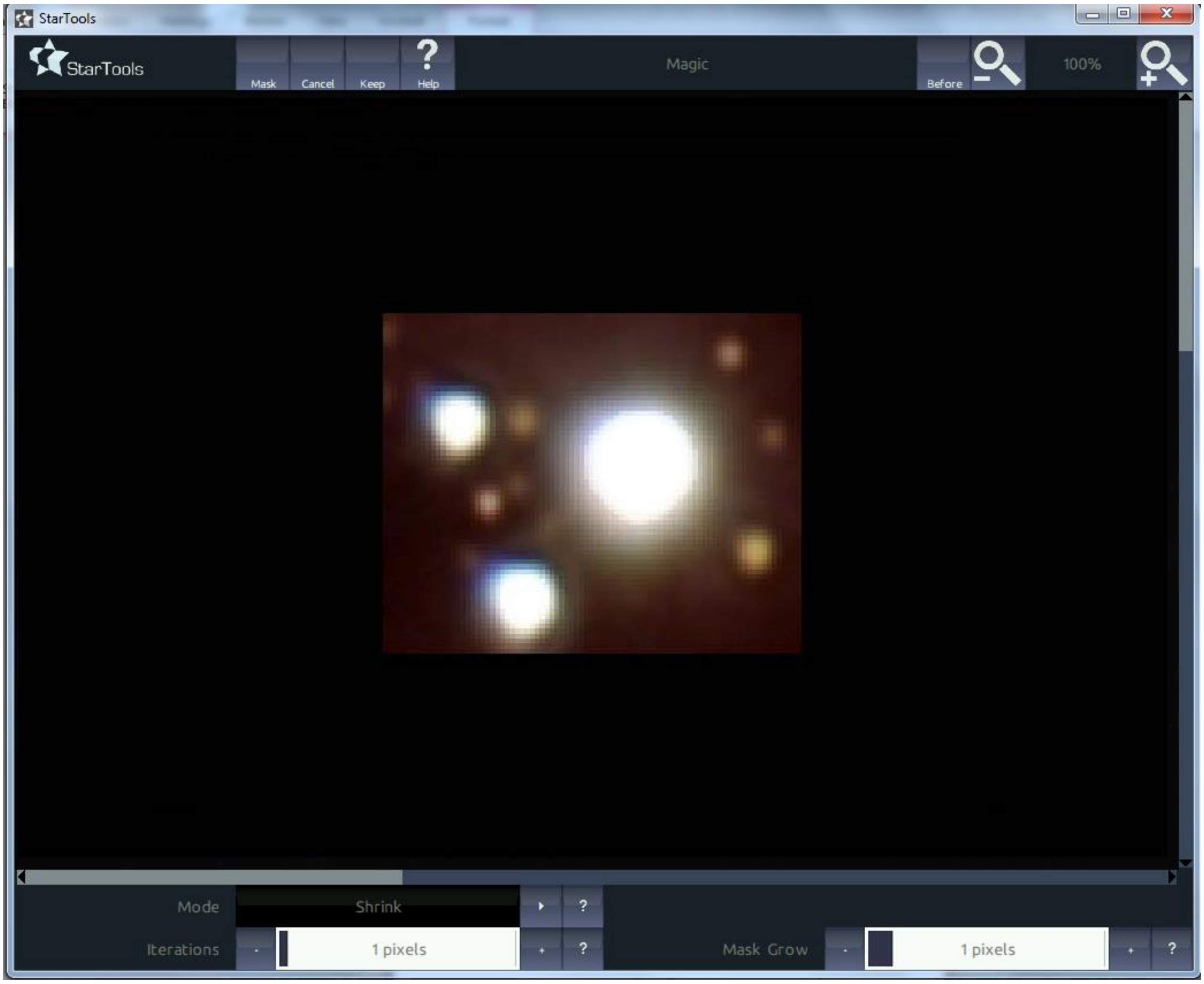


Illustration 41 - The 'Expand Core' operator in effect. Notice how the star has a more defined edge and the core is perfectly white.

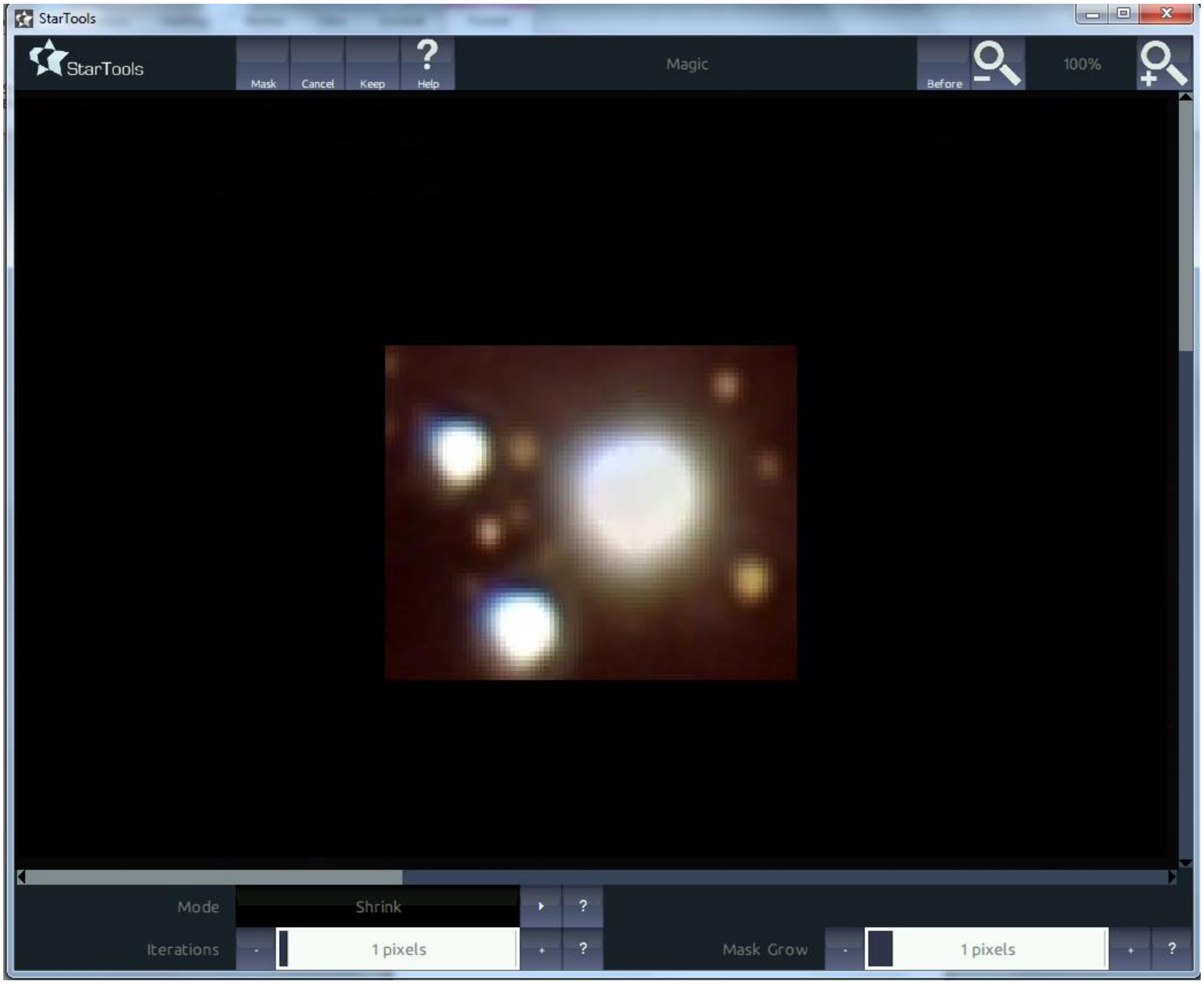


Illustration 42 - The 'Shrink Core' operator in effect. Notice how the star's core is no longer perfectly white, but instead has inherited the faint colouring of the outer rim of the star (often corresponding to the star's real colour).

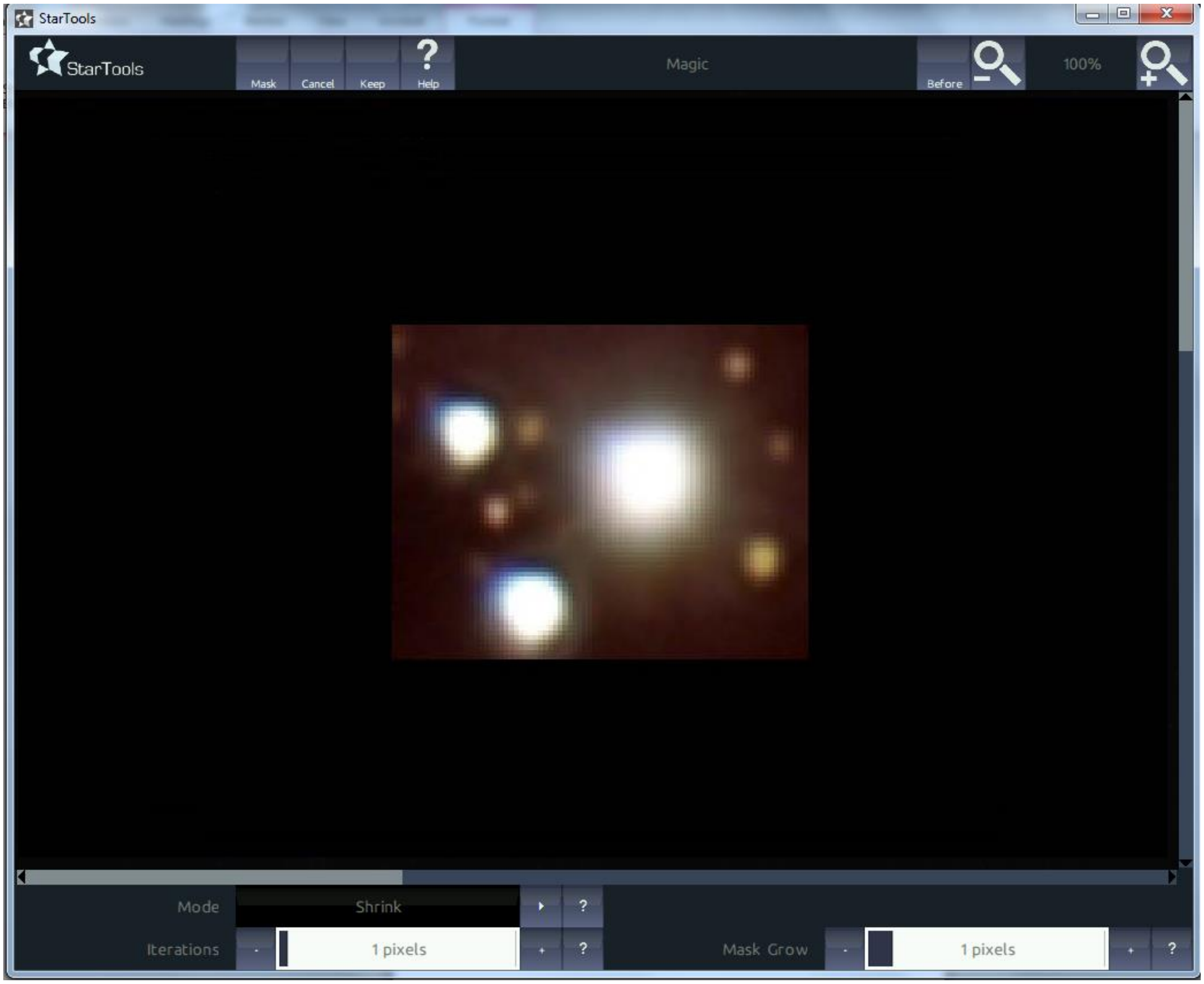


Illustration 43 - The 'Shrink' operator in effect. Notice how the star has shrunk dramatically.

Usage

The Magic module itself is fairly simple to operate with the following 3 parameters

- Mode - specifies what Magic should do with the star. There are 4 different algorithms to choose from;
- 'Shrink' - quite simply shrinks the star. An example can be seen in Illustration 43 where the star from Illustration 39 has been shrunk. Note that further processing (such as sharpening) may be appropriate.

- 'Shrink Core' - bleeds the edge color into the star's core, effectively colouring the core with the rim's colors. An example can be seen in Illustration 42 where the star from Illustration 39 has been subtly coloured according to the color at the edges of the star core (the latter often being the natural color of the star). The introduction of color information into the core now further enables you to colour the complete star in its natural colour by applying some increased 'Saturation' to the core in the 'Color' module.

- 'Expand Core' - expands the whiteness of the star core to the outer regions, making the star appear bigger and brighter, but also more defined. An example can be seen in Illustration 41 where the star from Illustration 39 has had its core 'expanded'. Note that the star may subsequently be shrunk as a whole to reduce the apparent size increase.

- 'Tighten/Debloom' - attempts to attenuate any glare or halo around a star. An example can be seen in Illustration 40 where the star from Illustration 39 has had its glare reduced.

- Iterations - specifies the amount of iterations the algorithm should go through. This parameter typically controls the effect that is selected by the 'Mode' parameter.

- Mask Grow - temporarily (and non-destructively) performs the specified amount of 'Grow' operations on the mask. Increase this value if artefacts, such as clearly visible 'rings' start to form.

How to make big/fat/white stars appear tighter

This question comes up a lot, so here is a quick guide on how to take care of stars that seem 'washed out' or have large halos.

To make big/fat/white stars appear tighter, there is a special 'tighten' setting in the Magic module. This module allows you to change general star appearance.

What you will want to do, is have StarTools create a mask for you that has those big/fat/white stars in them. This is very easy - in the mask editor, click 'Auto', 'Stars'. This preset will usually do a decent job at grabbing all the stars. However, this time, we're only interested in the big/fat/white stars. To grab only those, change the 'Selection Mode' into 'Highlights>Threshold'. This does exactly what it says on the tin - it grabs only very bright pixels that are bigger than the 'Threshold' parameter (which by default sits at 100%). All you need to do then, is set 'Threshold' to something smaller than 100% (let's say 90%). Now 'click' Do.

If all is well, you should notice that the cores of the big/fat/white stars are now 'green' in the mask editor (meaning they are now part of the mask). You might want to click 'Grow' once or twice to select neighbouring pixels that are also quite white.

Congratulations - you're now working with 'advanced masking techniques'!
Now click 'Keep' to keep your mask and to start using it in the module of your choice (the Magic module in this instance).

If you launch the Magic module with the mask we just created, you'll see it subtly shrinks the big/fat/white stars by default. Very useful for busy star fields that detract from a DSO. But it's not what you're after right now. To accomplish what you are after in this case, select the 'Tighten' setting for the 'Mode' parameter. What you'll notice now is that it gets rid of the halo around big/fat/white stars. It will only get rid of any pixels of halos that are part of your mask. Just so you don't have to go back to the mask editor to 'grow' your mask, for your convenience there is a 'Mask Grow' parameter that temporarily grows the mask, including more of those pesky halo pixels as you increase the setting.

Finally, the 'Iterations' parameter affects the 'strength' of the effect.

Filter

Overview

The FILTER module allows you to modify features in the image by the color by simply clicking on them. In effect, it can act as an after-the-fact artificial filter for your image, for everything from narrow(er) band filtering to acting like a fringe killer. FILTER can be used to bring out detail of a specific color (such as faint Ha, Hb, OIII, or S2 details) remove artefacts (such as halos) or isolate specific features.

By clicking on a location in the image, FILTER will sample the image at that location and determine the spectral line that corresponds to the sample. It will then apply the selected FILTER MODE to that spectral line.

TIP: Try using the FILTER module with a mask to only modify specific areas of the image.

Usage

[FILTER MODE] controls what FILTER should do once a spectral line has been identified.

> [CONSERVATIVE NUDGE] makes a spectral line more pronounced, but only if over exposure (e.g. clipping) can be avoided.
>
> [NUDGE] makes a spectral line more pronounced.
>
> [PASS] keeps a spectral line and reduces the rest of the spectrum.
>
> [REJECT] reduces a spectral line's intensity, while preserving the rest of the spectrum.
>
> [FRINGE KILLER] attempts to remove halos and fringes around stars that have a designated color. It is a very effective tool to remove the purple halos around stars that some of the more budget-friendly achromatic refractors produce. For best results, put the offending stars in a mask and click on the coloured star halos repeatedly in different spots to eliminate the fringes completely.

[SAMPLING MEHTOD] controls how a spectral line is sampled from the image.

[3x3 AVERAGE] samples a patch of 3 by 3 pixels and take the average.

[SINGLE PIXEL] only uses the pixel that was clicked as the sample to base the spectral line on.

[FILTER WIDTH] controls how much of neighboring parts of the color spectrum are influenced by the FILTER operation.

[MASK FUZZ] specifies the KERNEL RADIUS of an optional Gaussian blur, to be applied to the mask. If a mask is used to selectively sharpen the image, using this parameter will ensure smooth (and hence undetectable) transitions between sharpened and unsharpened parts of the image.

Example: Fringe and Halo Killer

If you acquire your data with an achromatic refractor, your data may be suffering from the purple halo effect (chromatic aberration). There is a very quick and easy way in StarTools to get rid of the purple/bluish fringes, from where you can further reduce any halos using the 'Filter' module. You can go from this;

Illustration 44 - M42 with halos

To this;

Illustration 45 - M-42 after using the Filter module

To attack the purple fringes, simply launch the 'Filter' module;

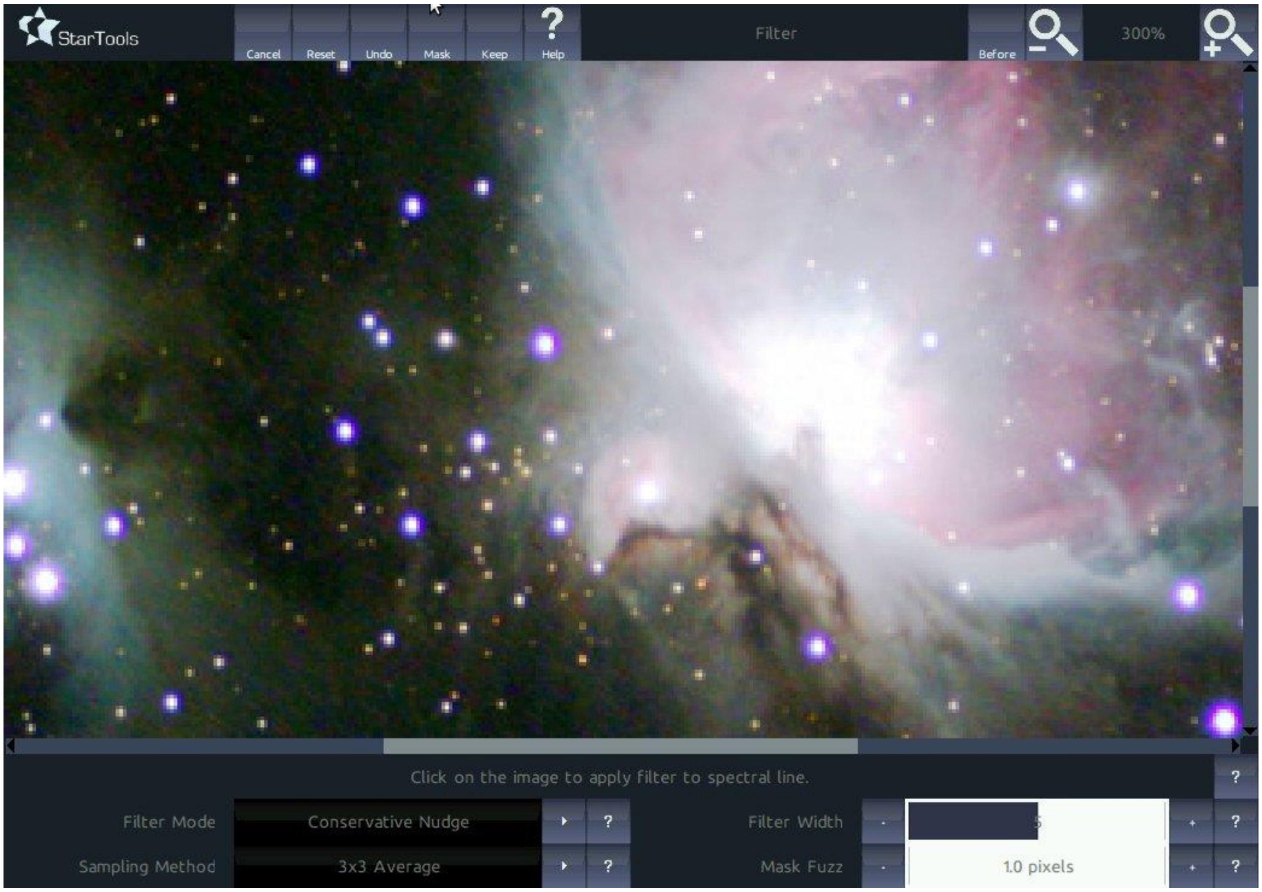


Illustration 46 - The Filter module

Click Mask, and create a star mask that is suitable. You'll want to include anything that is very bright and/or over exposed, i.e. anything that can cause the chromatic aberration to be prevalent;

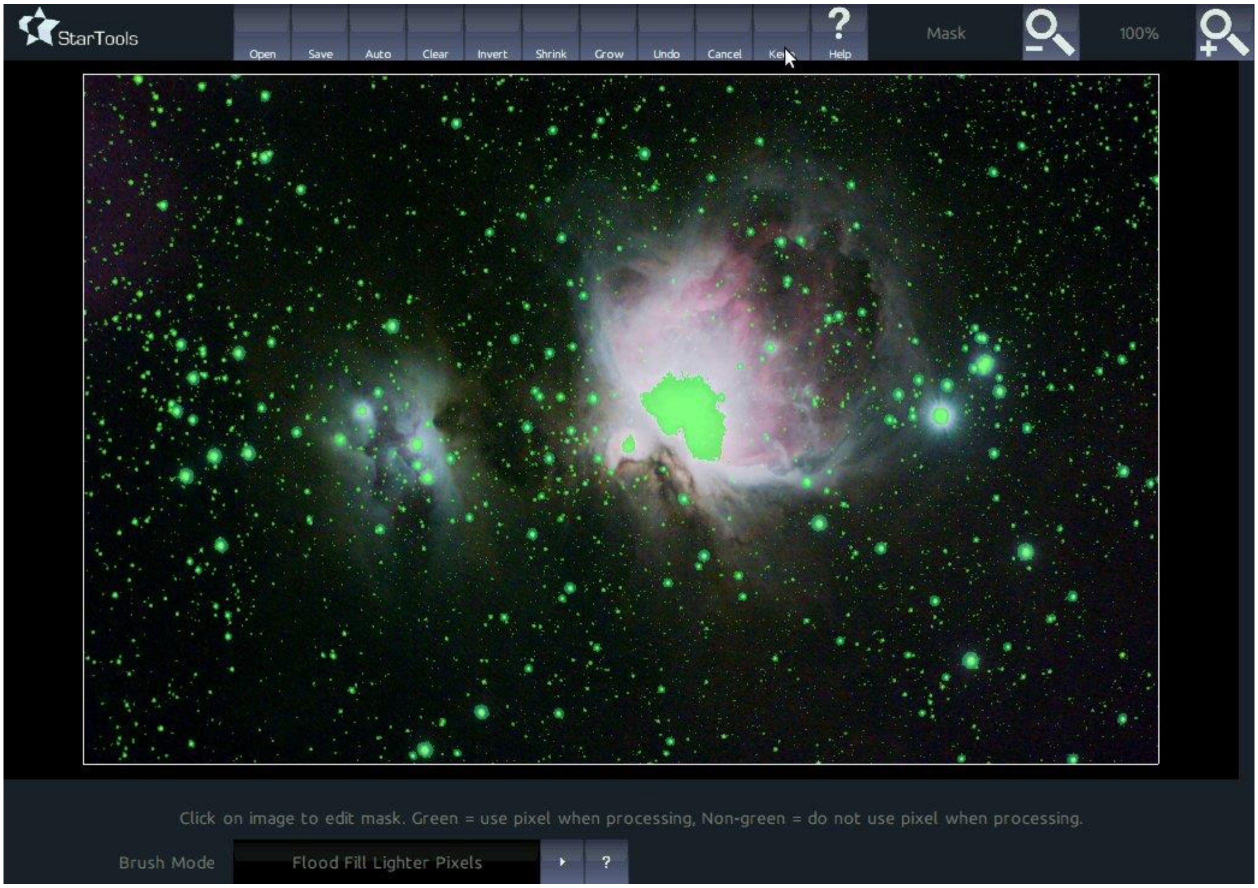


Illustration 47 - Creating the mask

Next, back in the 'Filter' module, set 'Filter Mode' to 'Fringe Killer' and 'Filter Width' to '1' (the latter lower value makes the filter more responsive to colours that are close - but not exactly the same - to the ones you click). All you have to do now is zoom into a star that exhibits the chromatic aberration and click on the offending halo.

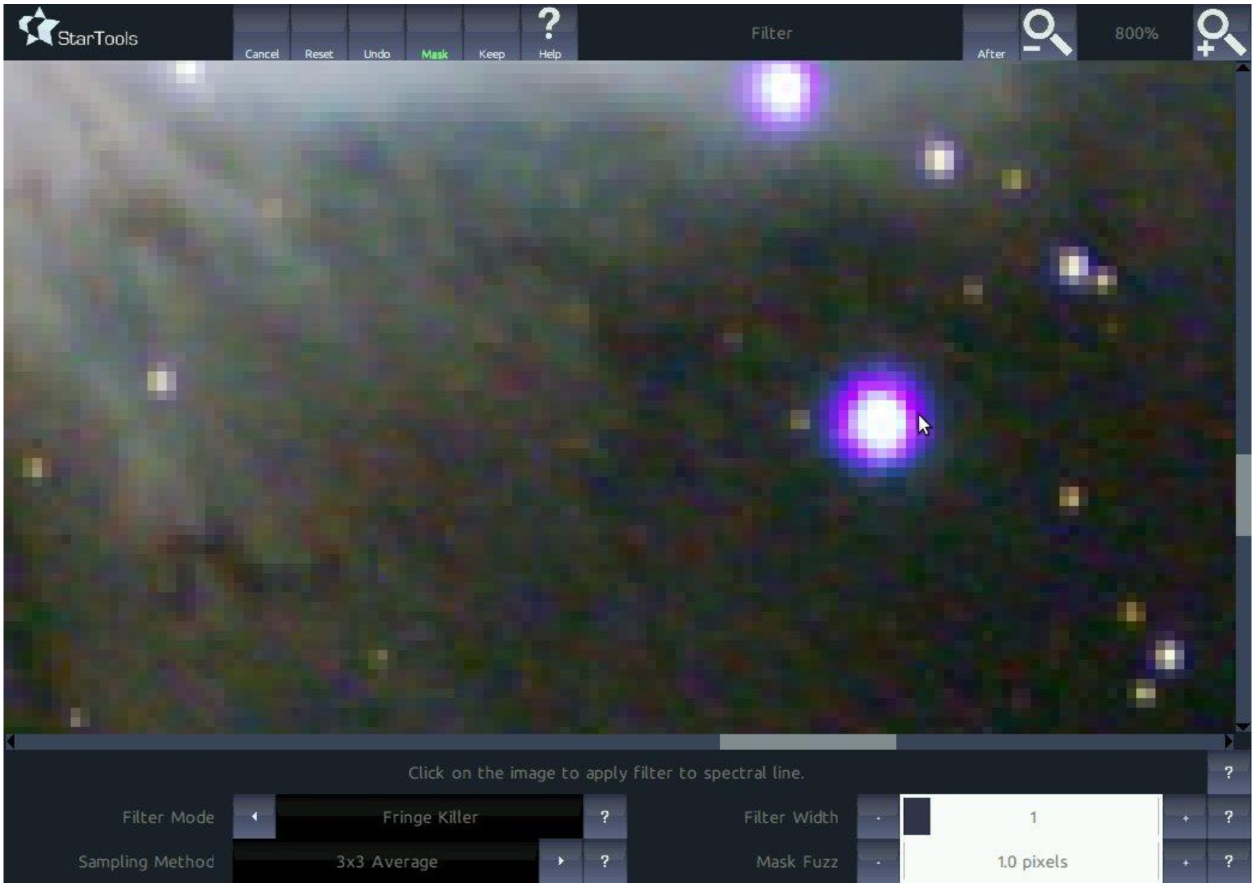


Illustration 48 - Zooming in on a star

After clicking a few times, this is the result;

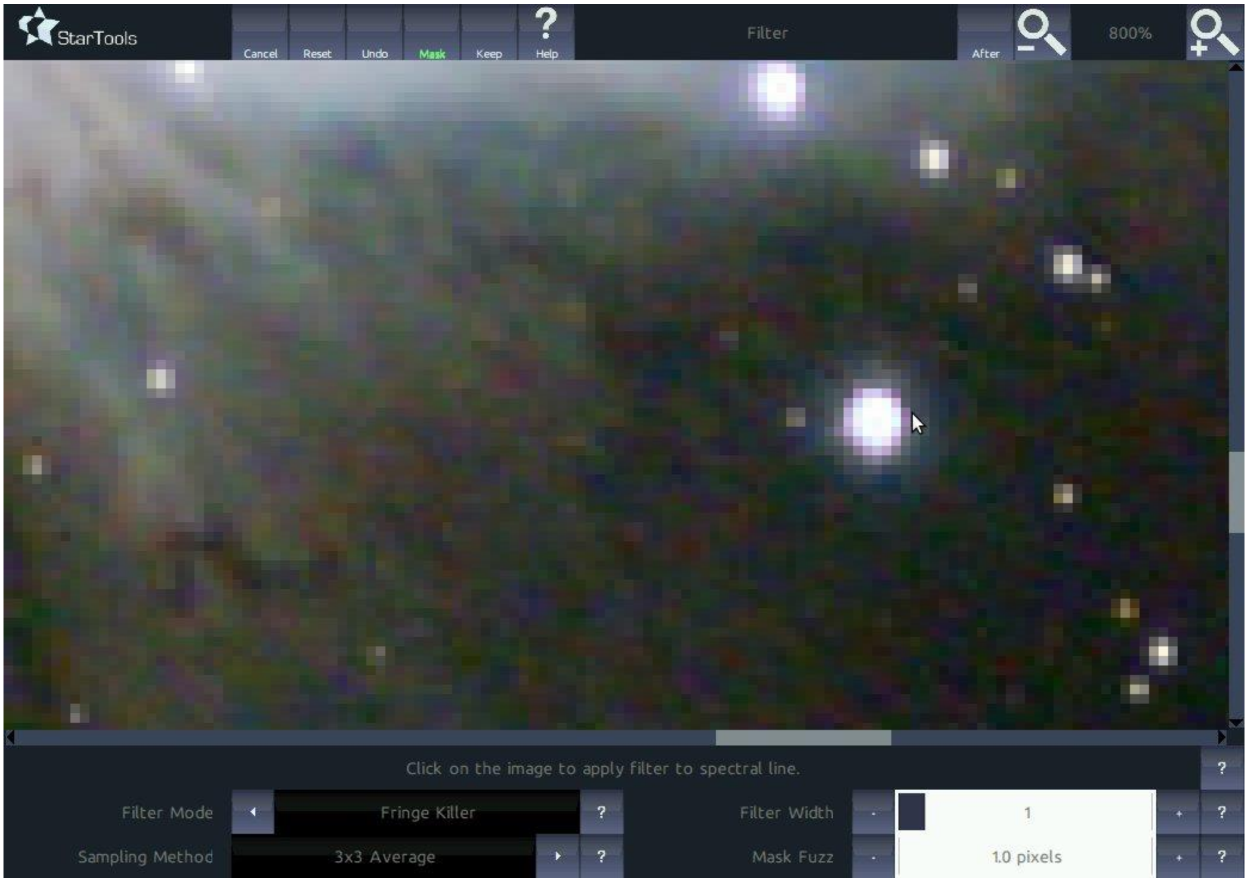


Illustration 49 - Result of using the Filter module

Layer

Overview

The Layer module allows the user to layer two separate images on top of each other by means of pixel math. Many additional algorithms and processing techniques may be emulated this way.

NOTE: In the LAYER module, the abbreviations ‘fg’ and ‘bg’ stand for ‘foreground’ and ‘background’ respectively.

TIP: After using any module in StarTools, you can layer the old image with the new image by clicking the ‘Undo > FG’ or ‘Un> BG’ buttons to copy the undo buffer (e.g. the old image) to the foreground or background respectively. Doing this gives you the after-the-fact control over a module’s output, so you can refine the result even more, for example by applying the automatically generated brightness (luminance) masks.

TIP: You can chain multiple operations by clicking the ‘Copy’ button to copy the layer result and then pasting the result into the background or foreground by using the ‘Paste >Bg’ or ‘Paste >Fg’ buttons respectively.

TIP: You can quickly integrate 2 exposure lengths (for example M42’s core and the rest of the nebula by selecting the ‘Max Contrast” or ‘Min Distance to 1/2 Unity’ for FILTER TYPE and increasing the FILTER KERNEL RADIUS until a smooth blend is achieved.

Usage

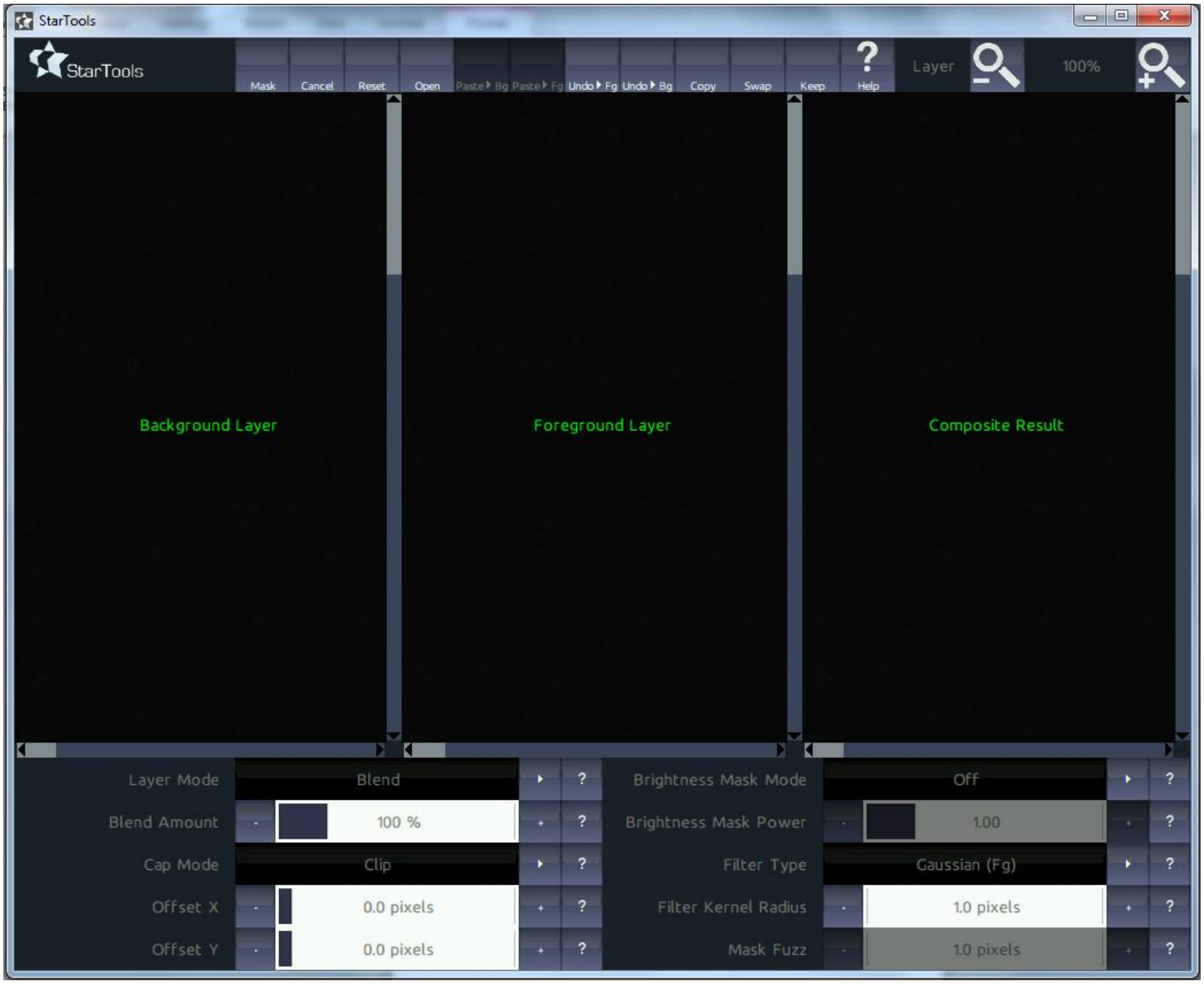


Illustration 50 – The LAYER module interface.

- [LAYER MODE] defines how the foreground image should be layered on top of the background image.
  - [BLEND] simply copies the foreground over the background. By varying the BLEND AMOUNT parameter you can make a blend of the background and foreground images.
  - [LIGHTEN] only copies those pixels that are lighter in the foreground over the background.

    - [DARKEN] only copies those pixels that are darker in the foreground over the background.
    - [MULTIPLY] multiplies the background image by the foreground image.
    - [ADD] adds the foreground image to the background image.
    - [SUBTRACT] subtracts the foreground image from the background image.
    - [DIFFERENCE] calculates the difference (positive or negative) between the foreground image and the background image.
    - [OVERLAY] overlays the foreground image on top of the background image. It darkens the image, but not as much as the MULTIPLY mode.

- [BLEND AMOUNT] controls how much of the foreground image is layered on top of the background image.
- [CAP MODE] defines how negative and over-unity values should be treated
    - [CLIP] truncates the negative values to 0 and over-unity values to unity.
    - [NORMALIZE] stretches the levels to span the whole available dynamic range.
- [OFFSET X] specifies the horizontal offset of the second layer relative to the first.
- [OFFSET Y] specifies the vertical offset of the second layer relative to the first.
- [BRIGHTNESS MASK MODE] specifies optional masking using the brightness information contained in both the foreground and the background. This feature allows you to blend an image based on brightness. Various modes are available.
    - Where fg is dark, use bg
    - Where fg is light, use bg
    - Where fg is light &dark, use bg
    - Where fg is grey, use bg

  - Where composite is dark, use bg
  - Where composite is light, use bg
  - Where composite is light & dark, use bg
  - Where composite is grey, use bg

- [BRIGHTNESS MASK POWER] specifies power that should be applied to a pixel in the brightness mask, giving you control over the range of the very brightest (or darkest) pixels that will still impact the brightness mask blending procedure.
- [FILTER TYPE] specifies the type of filter to be used on the foreground layer before layering is performed.
  - [GAUSSIAN (FG)] applies a Gaussian filter with a kernel size of FILTER KERNEL RADIUS to the foreground layer.
  - [MEDIAN(FG)] applies a median filter with a window size of (1 + [FILTER KERNEL RADIUS] x 2) to the foreground layer.
  - MEDIAN OF MEDIAN HALF (FG)] applies a 'mean of median half' filter with a window size of (1 + [FILTER KERNEL RADIUS] x 2) to the foreground layer.
  - [MINIMUM (FG)] applies a minimum filter with a window size of (1 + [FILTER KERNEL RADIUS] x 2) to the foreground layer.
  - [LIGHTNESS (FG)] applies a maximum and minimum filter with a window size of (1 + [FILTER KERNEL RADIUS] x 2) to the foreground layer. It then takes the mean of the minimum and maximum.
  - [DIFFERENTIAL ADAPTIVE NOISE] compares foreground and background image and suppresses any noise increase in the foreground due to brightening.
  - [MAX DISTANCE TO 1/2 UNITY] selects between foreground and background based on which value is closest to ½ unity (gray). Use FILTER KERNEL RADIUS to make a smooth blend between the two.

  - [MAX CONTRAST] selects between foreground and background based on which value adds the greatest contrast to the image. Use FILTER KERNEL RADIUS to make a smooth blend between the two.
  - [SOBEL] performs a Sobel edge detection operations on the foreground image.
  - [MEDIAM HORIZONTAL (FG)] performs a horizontal median filter with a horizontal kernel size of (1+ [FILTER KERNEL RADIUS] x 2) pixels.
  - [FRACTIONAL DIFFERENTATION] APPLIES Fractional Differentation Filtering on the foreground image (see http://arvix.org/abs/1005.4323). While not of immediate photographic use, this algorithm can demonstrate hidden structural detail otherwise hidden. FILTER KERNEL RADIUS governs the v parameter, while alpha is fixed to 0.5.

- [FILTER KERNEL RADIUS] specifies the kernel size of an optional filter, to be applied to the foreground layer before layering is performed.
- [MASK FUZZ] If a mask is used to selectively noise reduce the image, using the MASK FUZZ parameter will ensure smooth (and hence undetectable) transitions between background and foreground image. This option is unavailable if no mask is active (e.g. when all pixels in the whole image are tagged for processing).

  TECHNICAL EXPLANATION: This parameter specifies the kernel radius of an optional Gaussian blur, to be temporarily applied to the mask.

Example 1-Integrating frames of different exposure lengths

In this example we will demonstrate how to use the Layer module to integrate two frames that were recorded at different exposure lengths.

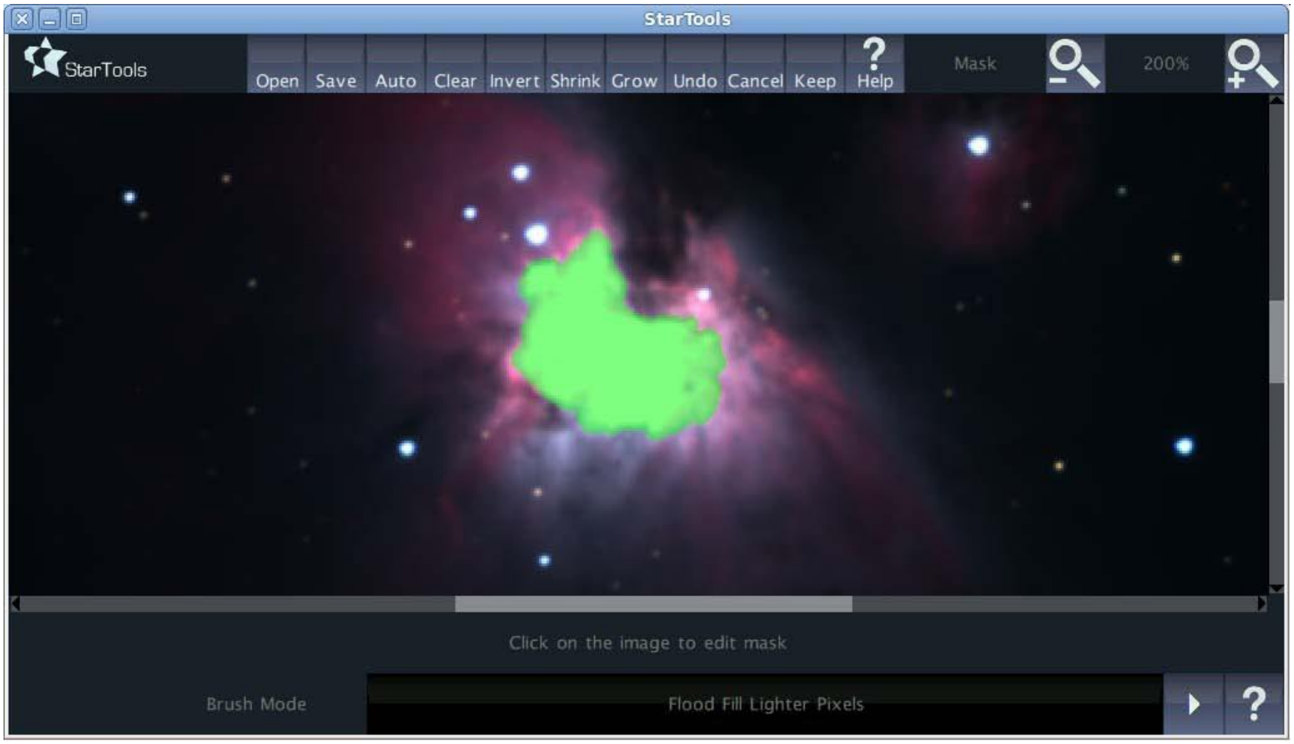


Illustration 51 - M42 with overexposed core selected as the Mask. Image courtesy of Rowland Cheshire.

We first start off with the longest exposed image. In Illustration 51 we have opened the Mask editor, cleared the old Mask, set the 'Brush Mode' to 'Flood Fill Lighter Pixels' and clicked just outside the overexposed core of M42. This has now selected the core.

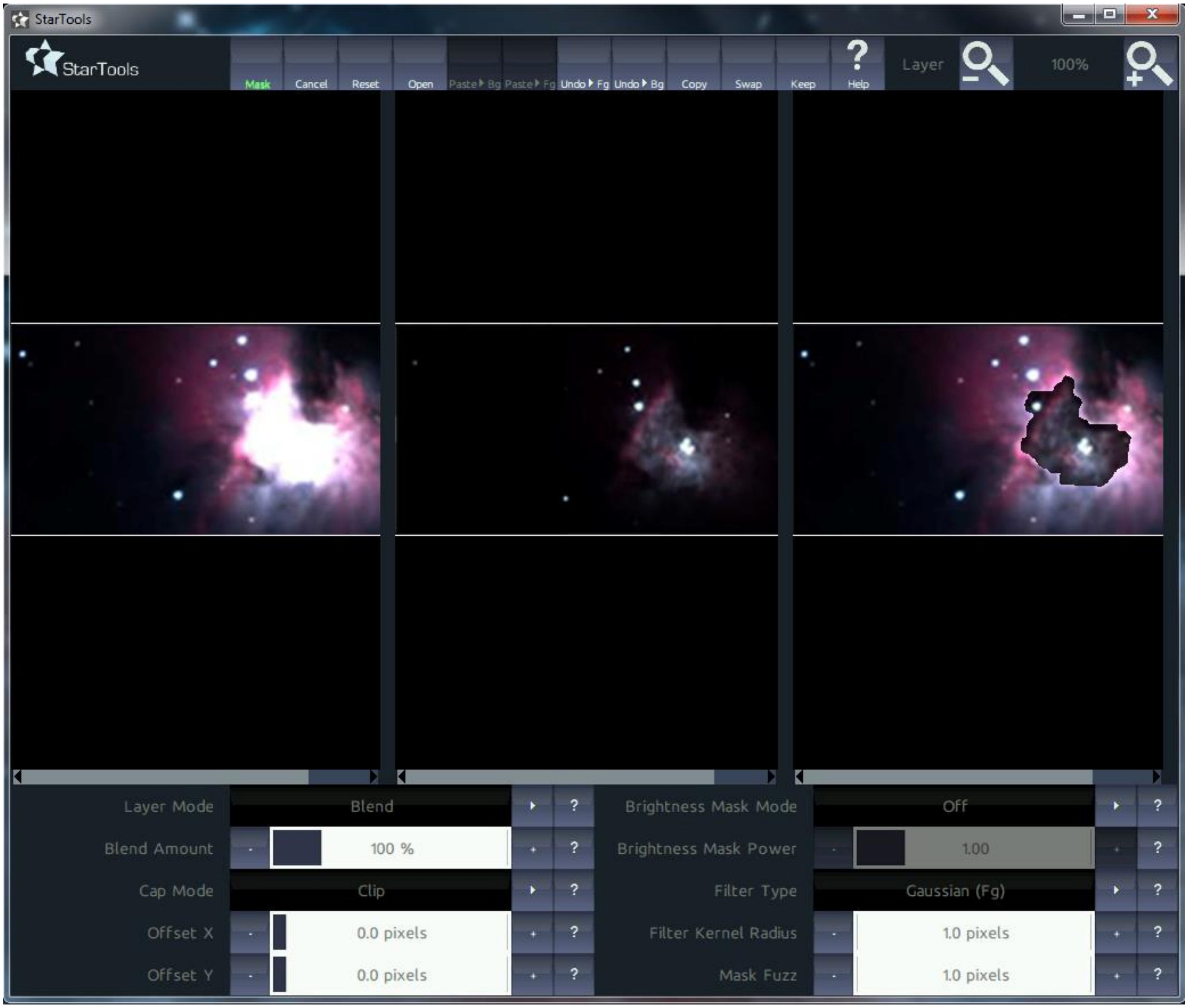


Illustration 52 -The two different exposures of M42 in the Layer module, composed according to the Mask from Illustration *51*.

Next, we open up the Layer module. We have loaded the less exposed image into the 2nd layer (i.e. the middle image) by clicking on the 'Open' button. The composed result (in the right image), as seen in is close to what we want, however there is clearly a transition between the two images at the Mask's boundaries. However, by simply using the 'Mask Fuzz' parameter we can make the two images transition smoothly as seen in Illustration 53.

Note that it may be worth experimenting with the 'Mask Fuzz' parameter and/or 'Grow'-ing the Mask in the Mask Editor, though, in this particular example, we did not need to.

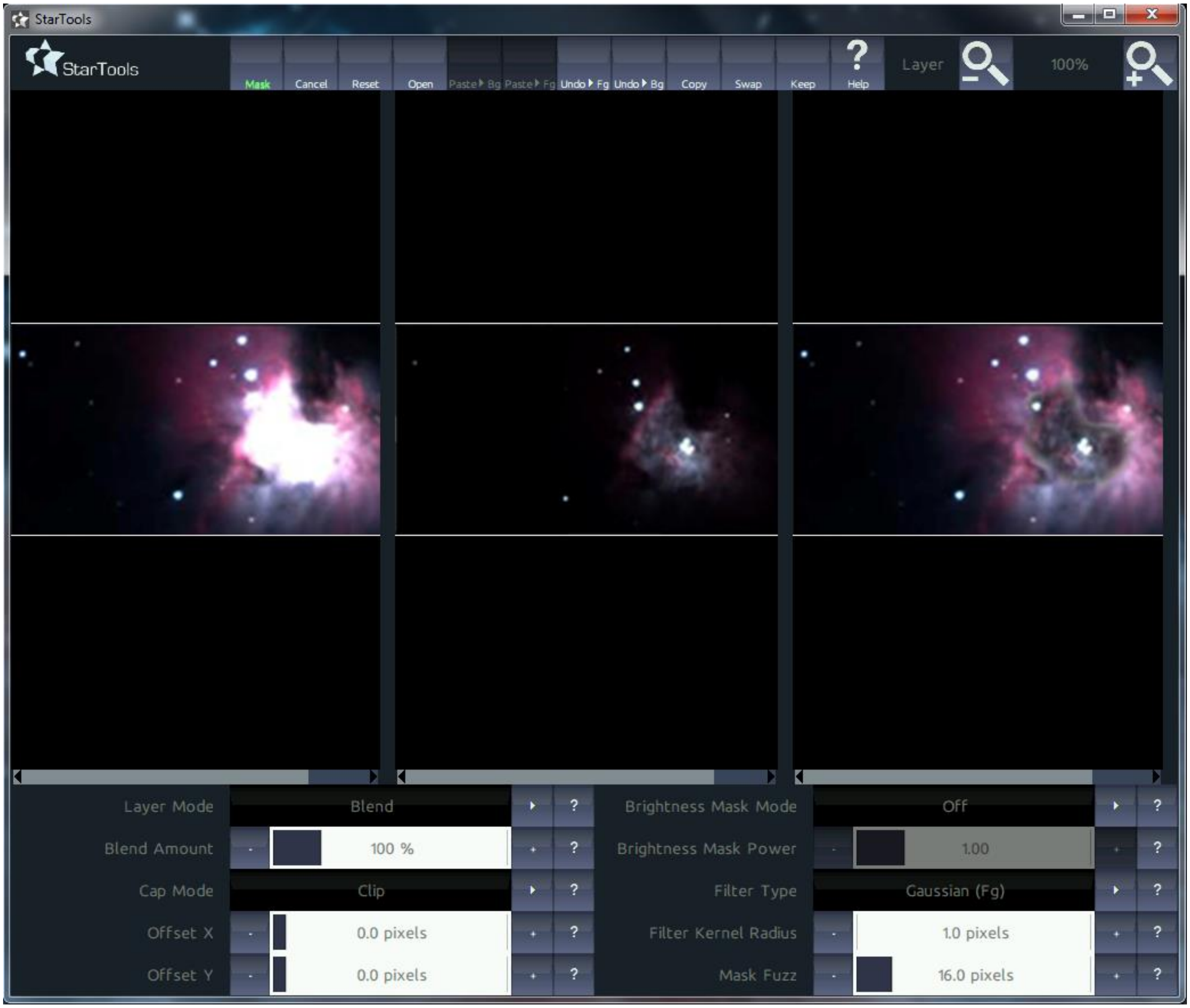


Illustration 53 - The two different exposures of M42 in the Layer module, with the 'Mask Fuzz' parameter set**.**

We now have successfully made a composite of two differently exposed frames.

Example 2-Rounding elongated stars

Elongated stars may have several causes. If your stars get progressively elongated towards the edge, this may be due to coma, which may be corrected by using the Lens module (see page80) or inserting a physical field flattener/coma corrector in your image train.

Sometimes, however, tracking issues may be the cause elongated stars and this will result in streaks pointing the same way. In this case, the Layer module offers a way to correct this with

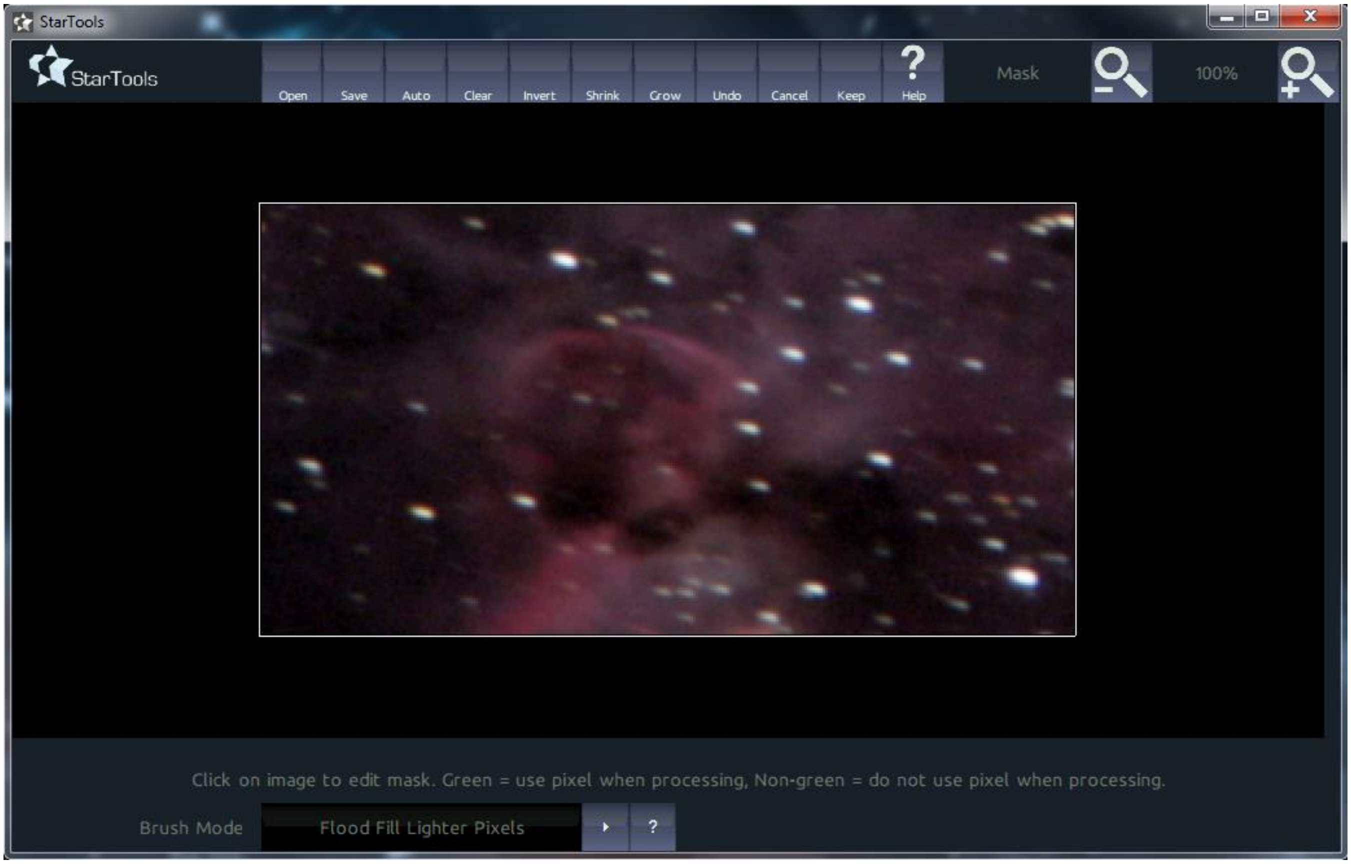


Illustration 54 - Keyhole Nebula with severe tracking issues

We open up an image of the Keyhole Nebula, which suffers from severe tracking issues. We then go into the Mask editor (Illustration 54). Note that we cleared the mask in Illustration 54 for clarity, but clearing the mask is not a necessary step.

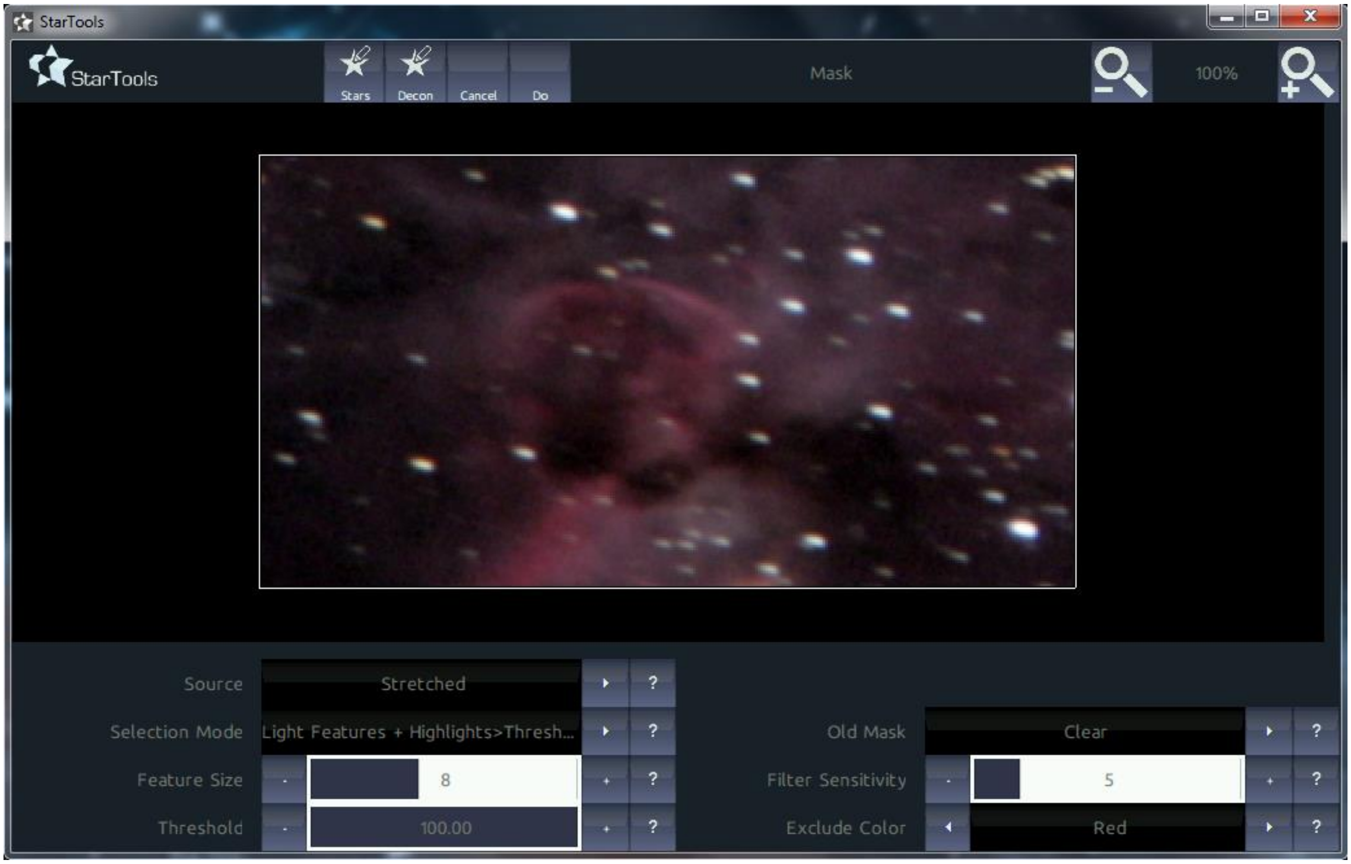


Illustration 55 - Using the 'Auto' feature to automatically select the stars.

We'd like to keep the Nebula intact and only modify the stars, so we use the 'Auto' feature and the ‘Stars’ button to automatically select the stars for us (see also the example on page 158) as can be seen in Illustration 55.

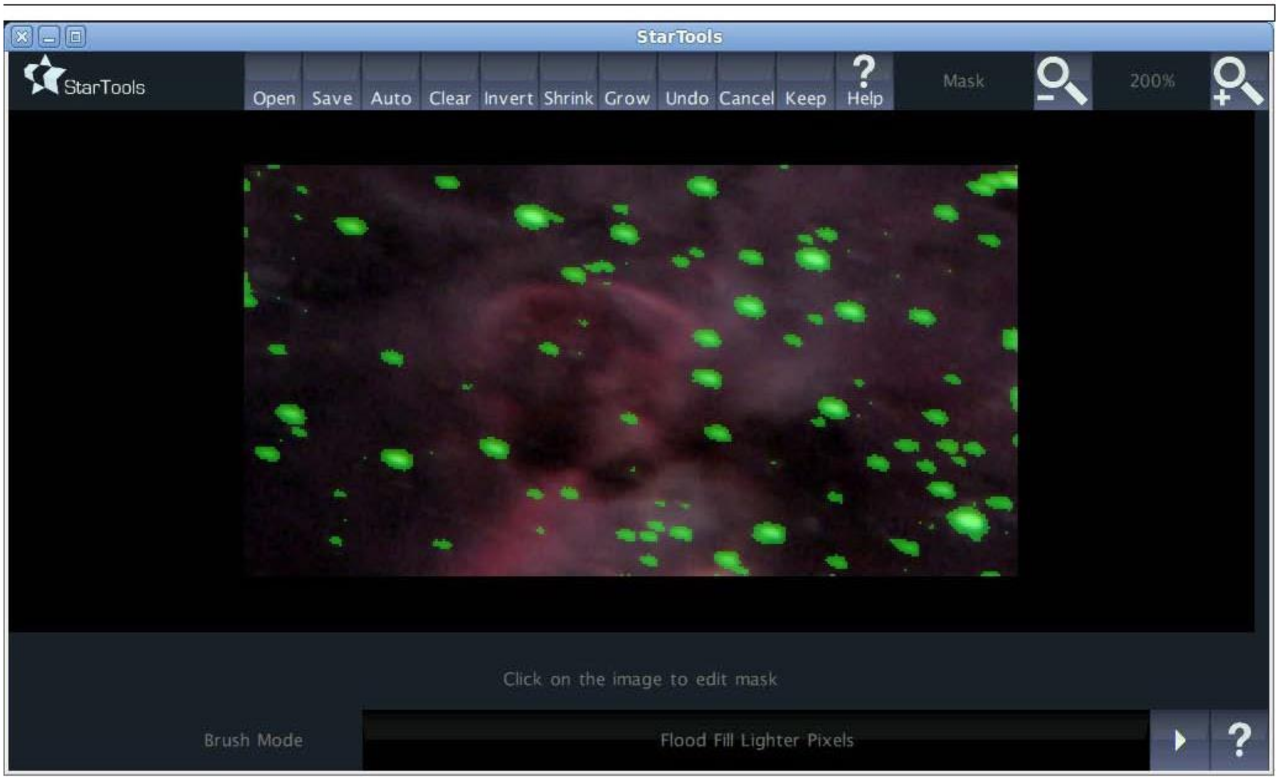


Illustration 56 - Keyhole Nebula with the elongated stars automatically selected.

The 'Auto' feature will return us to the Mask editor screen with the stars selected for us, similar to Illustration 56.

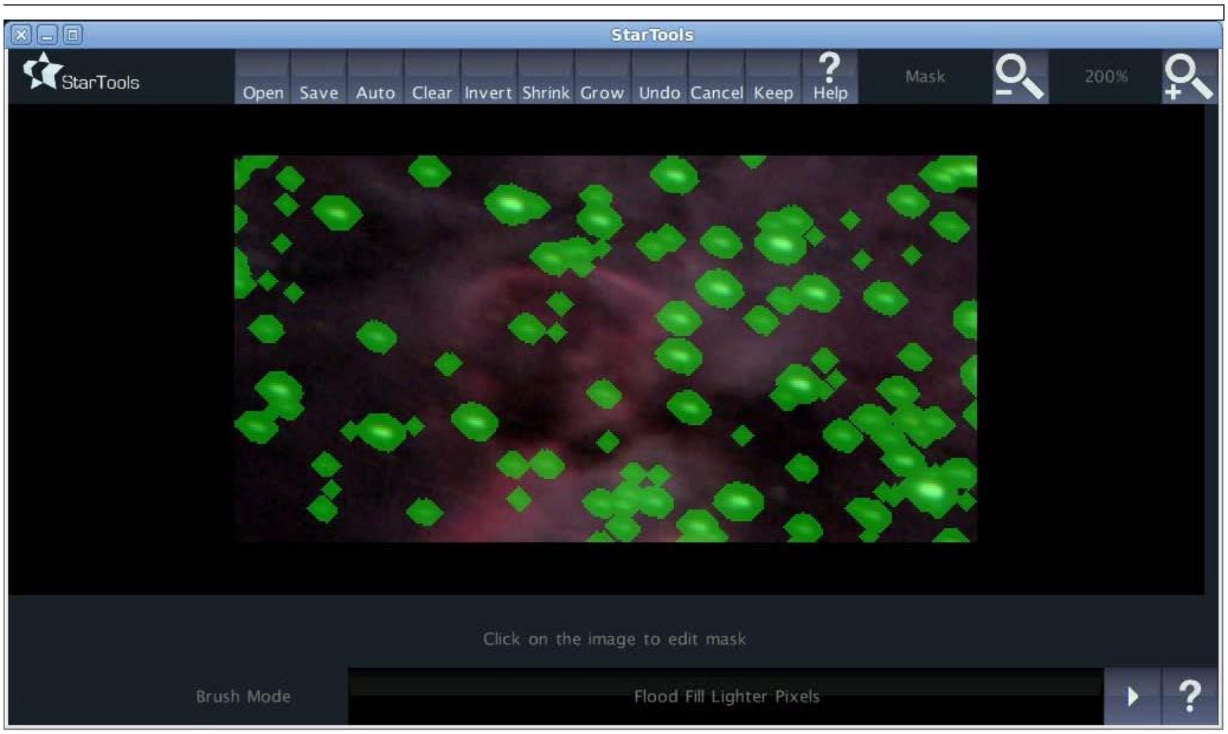


Illustration 57 - The result of clicking the 'Grow' button a few times.

However, for star rounding purposes we require a few more pixels around the stars in our Mask. Therefore we click the 'Grow' button a few times and end up with as screen similar to the one in Illustration 57.

Illustration 58 - Launching the Layer module with the Keyhole Nebula image loaded.

We now launch the Layer module as can be seen in Illustration 58.

Illustration 59 - The Keyhole Nebula in the Layer module

The Layer module display with default parameters showing no change initially to the stars in Illustration 59.

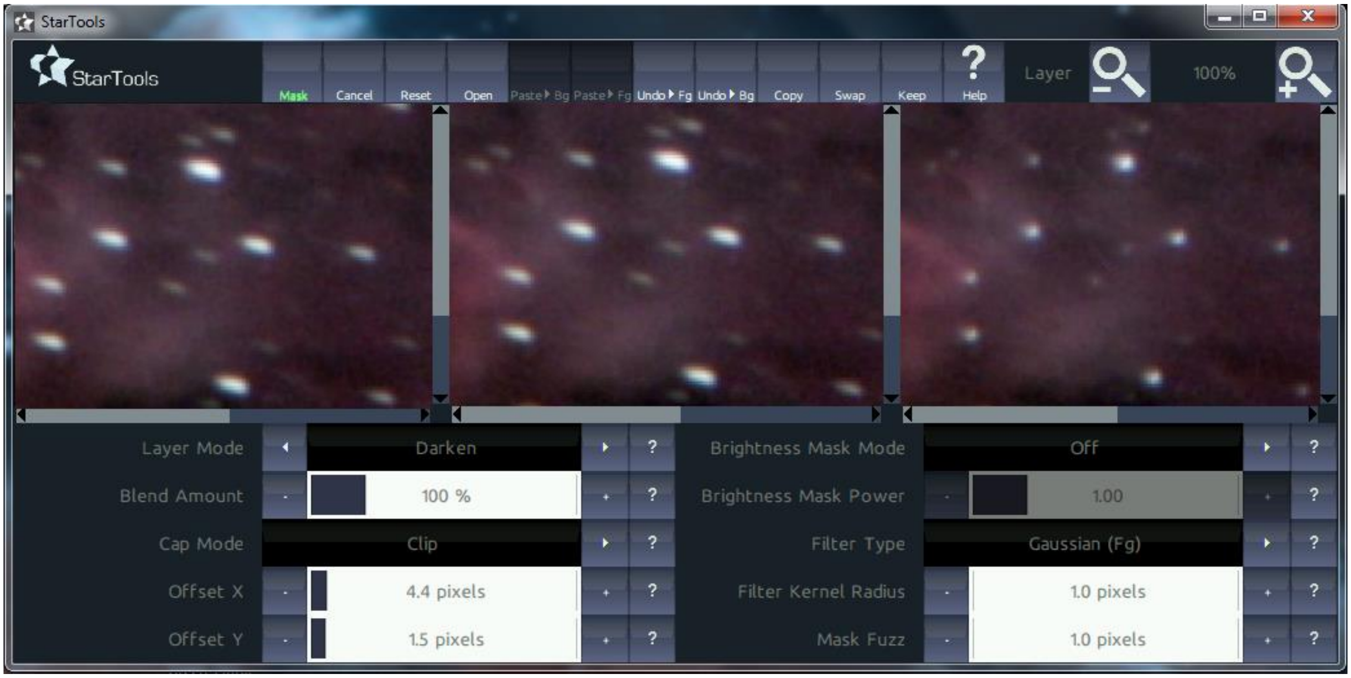


Illustration 60 - After setting 'Mode' to 'Darken' and adjusting 'Offset X' and 'Offset Y' we end up with nice round stars.

We now set the 'Layer Mode' to 'Darken'. We then keep adjusting the 'Offset X' and 'Offset Y' parameters until the stars look nice and round, similar to the screen in Illustration 60. We don't worry about any artefacts around the stars at this stage.

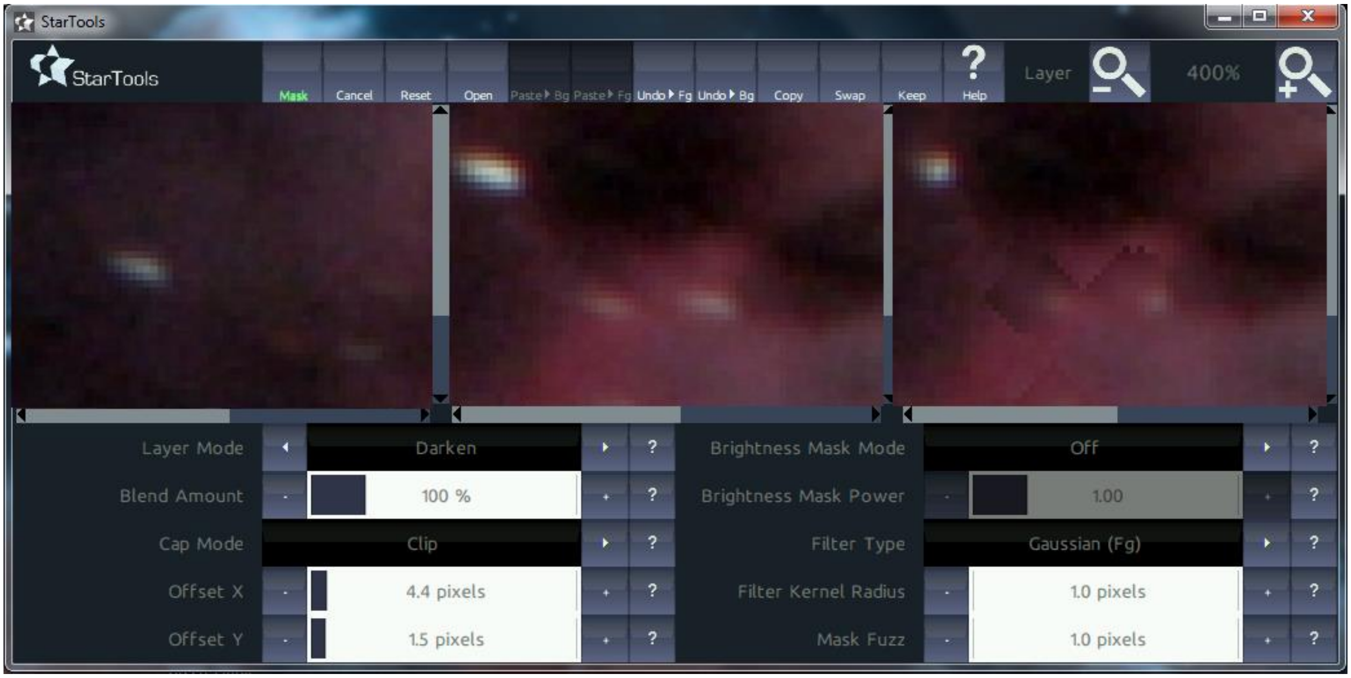


Illustration 61 - Artefacts around the 2 stars in the middle of the far right image.

In some case, the rounding procedure may have caused some artefacts in the form of sharp boundaries, as can be seen around the two stars in the middle of the right image of Illustration 61.

In this case, the 'Mask Fuzz' parameter comes to the rescue, eliminating the artefacts completely, as seen in Illustration 62. Note that these artefacts were also the reason we grew our initial star mask by a few pixels, so we had some additional pixels to blend with.

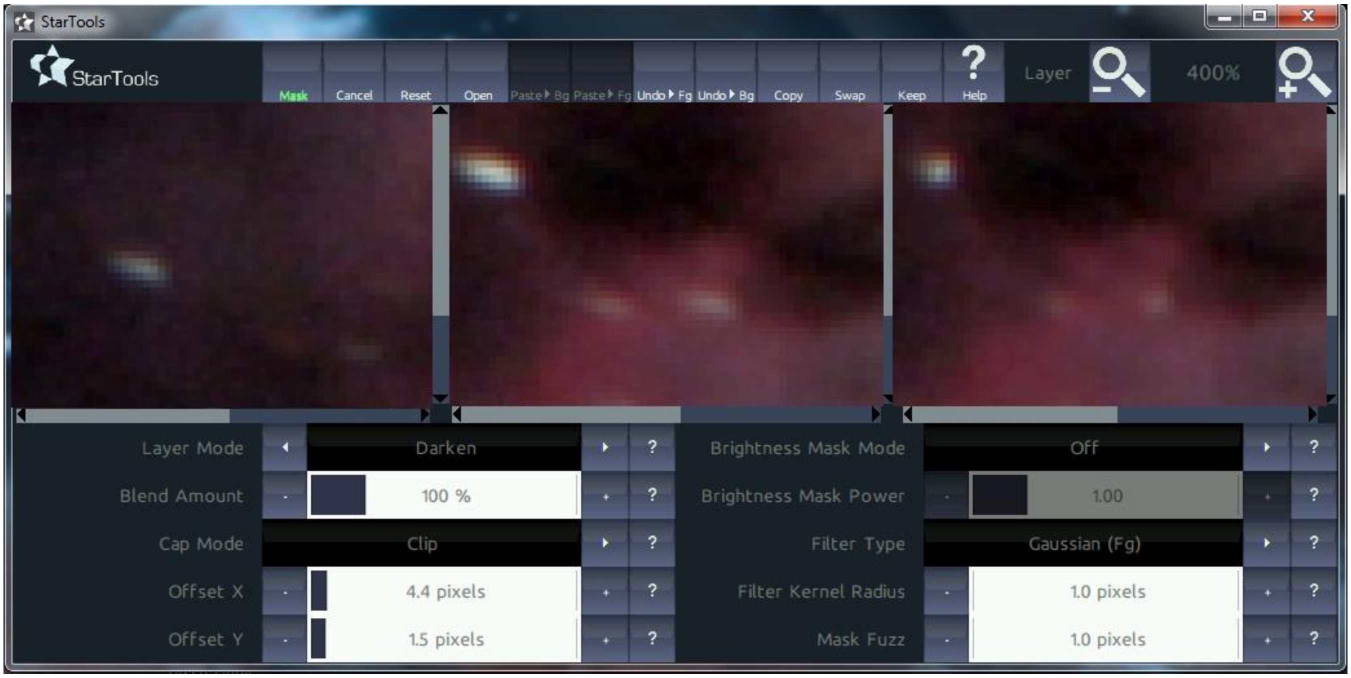


Illustration 62 - The 'Mask Fuzz' parameter has eliminated the artefacts around the middle two stars.

We should now have an image with perfectly round stars, with any background objects still intact.

Please note that this procedure relies on the fact that stars are bright dots that are brighter than their surroundings. It is therefore not possible to use this technique to enhance more complex objects. However, a technique that may potentially be useful for more complex cases is a technique called 'Deconvolution'.

DENOISE

Overview

[DENOISE] is a powerful de-noising tool, offering you a way to get rid of all different types of noise while preserving detail.

This first screen lets you choose the filter type you wish to use as a basis to reduce noise in your image. The second screen is where you customize the noise reduction.

Usage

Clicking the TRACK button brings up the following options:

- Stop tracking, do final noise reduction
- Stop tracking, no noise reduction

Clicking on 'Stop tracking, do final noise reduction' will invoke the Wavelet De-Noise module. The only option available on the first screen is FILTER TYPE.

- [FILTER TYPE] defines the noise filter you wish to use as a basis to perform noise reduction.
  - [Distance Weighted Outlier Rejection] combines characteristics of both MEDIAN and GAUSSIAN NOISE DIFFUSION and often yields very good results with superior detail retention.
  - [Gaussian Noise Diffusion] applies smoothing using Gaussian kernels (aka Gaussian 'blurs') at different sizes. It yields good results in the face of a lot of noise.
  - [Median] applies a median filter which is very effective when trying to preserve edges in low to moderate noise data.
- After choosing the FILTER TYPE click the Next button. When the Wavelet de-noise routine completes the following message displays:
  - Noise Floor

> If your image exhibits a clear noise floor beyond which signal is overwhelmed by read noise, then bring up the 'Read Noise Compensation' parameter until the noise disappears. The 'Read Noise Compensation' should correspond to the luminance value in the image where the noise floor starts. Try to minimize this value as much as possible, using other parameters as much as possible to control noise first.

This second de-noise screen is where you perform noise reduction to taste.

Bring up the various SCALE levels to see noise reduction take effect at the various noise grain sizes. If your image suffers from noise grain of a specific size, simply find the matching SCALE parameter and increase its level. If your image suffers from a mixture of noise grain sizes, push up all SCALE parameters' levels.

NOTE: The DENOISE module separates the image into color and brightness information. This allows you to control noise reduction in both the color domain and the brightness domain.

TIP: If there are any parts that you feel do not stand anything to gain by noise reduction, you are encouraged to create a mask in the mask editor, so that you can mask these parts out.

TIP: If your image suffers from color 'blotches' (caused by noise in the color domain), set BRIGHTNESS DETAIL LOSS to 0%. This will ensure only the color blotches will be addressed, while any brightness detail is preserved.

- [SMOOTHNESS] specifies how finer detail that was successfully correlated by the SCALE CORRELATION feature to a larger structure (such as a spiral arm in a galaxy) should be treated. Higher values smoothen this correlated detail, while lower values keep all of the correlated detail in the larger scale structure intact.
- [SCALE 1] controls the amount of fine noise reduction. This is the scale that governs the smallest noise, such as single pixel noise.
- [SCALE 2] controls the amount of medium to small noise reduction.
- [SCALE 3] controls the amount of medium noise reduction.
- [SCALE 4] controls the amount of medium to large noise reduction.

- [SCALE 5] controls the amount of noise reduction.

- [REDISTRIBUTION KERNEL] specifies what should be done with the excess energy of pixels that were smoothened. The REDISTRIBUTION KERNEL specifies over how large an area the energy is scattered. Smaller values tend to preserve larger scale detail a little better, but can lead to larger scale noise blotches forming if not dispersed over a large enough area.

- [READ NOISE COMPENSATION] specifies a threshold below which linear noise reduction no longer applies (otherwise a core assumption in the DENOISE module), due to signal becoming overwhelmed with noise (typically read noise). Bringing up READ NOISE COMPENSATION will gradually switch to an alternative specialized noise reduction routine for pixel values below the specified threshold. The READ NOISE COMPENSATION feature will help avoid a ‘mottled’ appearance in the dark background if this arises.

- [BRIGHTNESS DETAIL LOSS] specifies how strict the noise reduction should be when leaving detail in the brightness information of the image alone. Larger values will make the algorithm less strict, possibly leading to some detail loss in lieu of more aggressive noise reduction.

- [COLOR DETAIL LOSS] specifies how strict the noise reduction should be when leaving detail in the color information of the image alone (ex. star color). Larger values will make the algorithm less strict, possibly leading to some color loss in lieu of more aggressive color blotch noise reduction.

- [SCALE CORRELATION] specifies how strongly detail in smaller scales correlates to detail in larger scales. For example, if a spiral arm in a galaxy is designed to be ‘real detail’ then the detail within the spiral arm is more likely to be real detail as well. The SCAL CORRELATION controls how far down this assumption propagates to the smaller detail levels.

- The SMOOTHNESS parameter controls how any detail that is deduced in this way is smoothened.

- [MASK FUZZ] controls a smooth blend between masked and non-masked parts of the image. If a mask is used to selectively noise reduce the image (not recommended when denoising with Tracking information available), using the MASK FUZZ parameter will ensure smooth (and hence undetectable) transitions between noise reduced and non-noise reduced parts of the image. This option is unavailable if no mask is active (e.g. when all pixels in the whole image are tagged for processing).
- This parameter specifies the kernel radius of an optional Gaussian blur, to be temporarily applied to the mask.

Denoise Preview Only Mode

With Tracking enabled StarTools Denoise module operates in 'Preview only mode' since StarTools is still tracking noise propagation in your image. Noise reduction is most effective when you are done stretching your image. Therefore you are given the opportunity to perform final noise reduction when you indicate you are done stretching by switching off tracking (by clicking the 'Track' button). In the meantime, you can use the Denoise module in preview-only mode to give you an idea of how well StarTools has been able to target noise up until now. The 'Keep' button will, however, be disabled.

BAND

Overview

The BAND module attempts to remove banding (‘stripes’) caused by sensor read-noise. The BAND module works best on linear (non-stretched) data, however the algorithms are adaptive and will attempt to compensate for any non-linearity.

Usage

- [ORIENTATION] specifies whether BAND should remove vertical or horizontal banding.
- [ALGORITHM] specifies which algorithm to use. [ALGORITHM 1] usually yields best results, but on occasion [ALGORTHM 2] may outperform it.

Repair

Overview

REPAIR attempts to detect and automatically repair star defects. Repair is useful to correct the appearance of stars which have been adversely affected by blooming, guiding errors, incorrect polar alignment, collimation issues or mirror defects such as astigmatism.

The REPAIR operation only yields usable results with a proper star mask set. To create a star mask, in the Mask editor, click 'Auto' and 'Do'.

TIP: For best results, make sure that all stars are separated in the mask by at least one 'off' pixel.

Usage

[ALGORITHM] specifies the algorithm that REPAIR should use while repairing stars.

> [WARP] founds stars by warping them into shape. This algorithm is particularly suitable for slight elongations due to bad tracking or mild coma, as well as any other small aberrations that leave a star oval.
>
> The [REDISTRIBUTE, CORE IS AVG LOCATION] algorithm attempts to redistribute all starlight pixels. This algorithm is suitable for severs aberrations and heavy coma that leaves starlight scattered, out of focus, (donut shaped) or otherwise severely distorted. When using this algorithm, please make sure all stars are separated in the star mask by at least one pixel. It attempts to reconstruct the original (core) location of the star by averaging all sample locations.
>
> The [REDISRIBUTE, CORE IS BRIGHTEST PIXEL LOCATION] algorithm is similar to REDISTRIBUTE, CORE IS AVG LOCATION algorithm but uses the brightest pixel it can find as the corrected star's core.
>
> The [DEBLOOM (VERTICAL STREAKS)] algorithm attempts to recover stars that were affected by CCD blooming. Note that this algorithm assumes that the blooming artefacts (e.g. streaks) occur in a straight vertical direction with no/minimal rotation present in the image.

[GROW MASK] specifies by how much the mask is temporarily grown while healing the background pixels of a star. Use a higher value if artefacts appear around corrected stars.

[RADIAL SAMPLES] if the WARP algorithm is active, this parameter specifies the amount of samples that will be taken when determining the ‘roundness’ of a star. A higher value is more likely to correct severely distorted stars, but may take longer to calculate.

[SUB SAMPLING] if the WARP algorithm is active, this parameter specifies the amount of sub samples per pixel to use while creating the RADIAL SAMPLES. A higher value may take longer to calculate.

How to make “eggy” stars appear rounder

This question comes up a lot as well, so here is a quick guide on how to make your stars appear rounder in StarTools.

You, again, will need a star mask. This time you'll be interested in all the stars (not just the big/fat/white ones). The 'Stars' preset in the Auto feature of the Mask Editor should help. If you find it is selecting too much stuff that aren't stars, increase the Filter Sensitivity setting. No one image is the same, so just experiment a little. If you have a DSO in your image that is predominantly red or blue, you could optionally instruct the Auto Mask Generator to ignore any red or blue detail by selecting 'Purple (Red + Blue) for the 'Exclude Color' parameter.

Once you're happy with your mask, run the 'Repair' module. This module will help you round your stars.

There are two algorithms that allow you to round your stars. One is the 'Warp' algorithm, which tries to warp your stars back into shape (much like a sculptor moulds a piece of clay). The other one is for heavily distorted stars that look more like patches of light without a distinct core than recognisable stars. The latter is called the 'Redistribute' algorithm. See which one gives you the best result. For both of them to work optimally, the stars in your mask *must* be at least one pixel apart. So if two stars overlap, make sure you 'erase' a single pixel line between them. If you don't do this, you may notice the two stars merge into one big star.

LIFE

Overview

Brings back life into an image by remodeling uniform light diffraction. Throughout the various processing stages, light diffraction (a subtle 'glow' of very bright objects due to lens or mirror diffraction) may be distorted and suppressed through the various ways dynamic range is manipulated. The LIFE module attempts to restore light diffraction uniformly throughout a processed image, imparting a natural sense of depth and ambience to an image that was otherwise lost. Usage of the LIFE module is only recommended as one of the last processing steps.

In addition to adding life to an image, the LIFE module can also be used to lift objects (such as DSOs) from very noisy data (the 'Isolate' preset).Creating a mask that roughly outlines the object to be lifted can further improve results.

Usage

- [Inherit Brightness, Color] optionally keeps brightness or color information intact.
- [Glow Threshold] specifies how bright a pixel needs to be in the original to be considered for diffraction. Lower values will result in more pixels being processed, also leading to longer processing times.
- [Output Glow Only] specifies if LIFE should output only the diffracted light calculated.
- [Saturation] specifies the color saturation level of the extracted glow before it is composited with the original image.
- [Detail Preservation] selects the way detail should be preserved in parts of the image that may be brightened.
    - [OFF] does not preserve any detail.
    - [MIN DISTANCE to ½ UNITY] uses the pixel that is closest to half unity.
    - [MAX CONTRAST] chooses whatever pixel maximizes contrast.

   - [LINEAR BRIGHTNESS MASK] uses a brightness mask that progressively masks out brighter values and uses the original values instead. Only pixels that are darker than the original image are kept.

- [Detail Preservation Radius] specifies a filter radius that is used for smoothly integrating processed and non-processed pixels, according to the algorithm specified with the DETAIL PRESERVATION parameter. Note that is parameter only takes effect when DETAIL PRESERVATION is not 'off'. It is grayed out otherwise.

- [Compositing Algorithm] specifies how LIFE should composite the original image and the calculated diffracted light.

   - [Power of Inverse]
   - [Multiply, Gamma Correct]
   - [Multiply, 2x Gamma Correct]
- [STRENGTH] defines the overall strength of the LIFE module

- [AIRY DISK SAMPLING] specifies the amount of samples in the Airy disk point spread function. Larger values yield a more accurate simulation, however will take longer to process.

- [AIRY DISK RADIUS] specifies the radius of the Airy disk point spread function that is used to diffract the light. Subtle diffraction rings may be visible around stars after processing. While this is a natural effect (first observed by George Biddell Airy), the frequency of their occurrence may be throttled by this parameter.

- [MASK FUZZ] specifies the kernel radius of an optional Gaussian blur, to be applied to the mask. If a mask is used to selectively 'ad life', using this parameter will ensure smooth (and hence undetectable) transitions between livened and non-livened parts of the image.

FLUX

Overview

FRACTAL FLUX offers a novel way of analyzing, enhancing and introducing detail into an image. It uses self-similarity (also known as fractals) to lift out or even create ‘interesting’ features. The automated identification of interesting features allows for a host of content aware operations, such as autonomous sharpening, feature augmentation and/or noise reduction.

Fractal flux may be used to add natural looking detail to an image, using detail already present as a basis. Do note that using this module in this manner adds detail that, though natural looking, is artificial in origin. Please also note that using fractal flux in this manner also tends to distort stars and/or any halos around them. It is therefore advisable to mask stars out if you intend to use the module in this way.

The nature of fractal flux is that its arisal strongly correlates with “uniqueness’ of features in an image. It is therefore also a good indicator of the presence of ‘interesting’ detail. By using fractal flux to modulate an Unsharp Mask algorithm, automated sharpening may be accomplished of ‘interesting’ features.

Usage

First Screen

- [WAVELET LIBRARY] specifies the amount of ‘knowledge’ the FRACTAL FLUX module starts off with. The smaller the amount of knowledge, the more flux may arise.

Second Screen

Offers Sharpen, Detail and Noise presets buttons at top.

- [ALGORITHM] selects the filtering algorithm to use.
    - [ADD FLUX] will add any flux to the image. This mode is particularly suited for introducing natural looking detail into an image that otherwise lacks any detail.

Please note that any detail introduced is nothing more than a ‘best guess’ based on the area’s similarity to other areas and any detail already present.

- [MODULATE UNSHARP MASK] uses the presence of flux to turn on and off an Unsharp Mask sharpening algorithm. This mode effectively performs localized sharpening of only ‘interesting’ features, while leaving others (including noise) alone.

- [POSITIVE FLUX] controls the amount by which any positive flux is amplified.
- [NEGATIVE FLUX] controls the amount by which any negative flux is amplified.
- [DETAIL FILTER] specifies the minimum allowable size of any flux concentrations. Increasing this value allows more and more detailed flux to pass through.
- [BRIGHTNESS MASK POWER] LUMA POWER specifies the power that should be applied to a pixel in the luminance mask.
- [BRIGHTNESS MASK MODE] LUMA MASK specifies optional masking using the luminance information.
- [FILTER AMOUNT] controls the strength of the Unsharp Mask sharpening.
- [FILTER RADIUS] specifies the maximum size of any details to be sharpened by the Unsharp Mask algorithm, or the filter width for other filters. The larger the radius, the larger the details that are sharpened.
- [FILTER FUZZ] specifies the kernel radius of an optional Gaussian blur, to be applied to the flux before it is passed to the Unsharp Mask routine for modulation. Increasing this value effectively ‘smears out’ the flux over a larger area so that detail around the flux is now also sharpened. Note that, because the flux is now being spread out, its intensity becomes lower. You may compensate for this by increasing the flux multipliers or by increasing the FILTER AMOUNT (sharpening strength) parameter.
- [MASK FUZZ] specifies the kernel radius of an optional Gaussian blur, to be applied to the mask. If a mask is used, using this parameter will ensure smooth (and hence undetectable) transitions between modified and unmodified parts of the image.

Color

Overview

The Color module allows for flexible manipulation of an image's colour, whether it be locally (through the use of a mask) or globally. Note that it is better to correct any aberrant issues (such as those caused by light pollution and gradients) using the WIPE module first. It is better to try and understand what is causing any incorrect coloring and eliminate the root cause. The COLOR module should only be used as a last resort in these cases.

When color can’t be recovered using the WIPE module, the color module offers you various ways of recovering correct color. The COLOR module can calibrate against a masked part of your image, for example, a galaxy, G2V star or a star field with a good mix of star temperatures, and apply the found RGB ratio to your whole image. To do so, create a mask with the object(s) to calibrate against and click the ‘Sample’ button. The COLOR module will perform the calibration and set the correct RGB ratios. Next ‘Clear’ and ‘Invert’ the mask (selecting all pixels) and return to the COLOR module. The ratios that were determined previously will now be applied to the full image, completing color calibration.

If you do not own a color calibrated monitor, a special viewing mode in the COLOR module may be of help. The ‘MaxRGB’ button will show you which color channel (red, green, blue) is the most dominant for each pixel. It is an objective measure for how your image is colored. This can help you fine tune your color and avoid discoloring your data due to an incorrectly calibrated monitor.

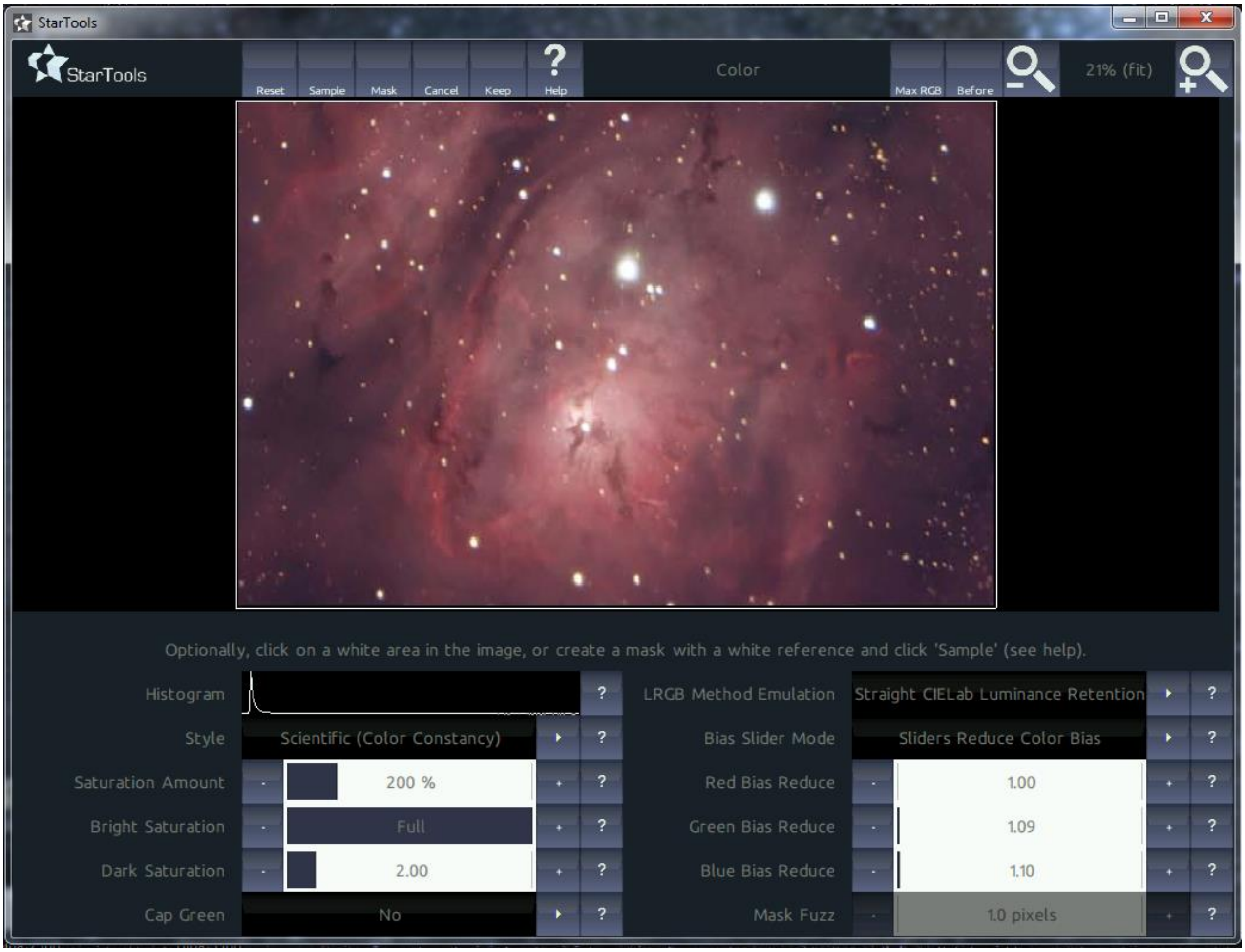


Illustration 63 - Color module interface.

Usage

Clicking on the COLOR module button displays the following:

> 'Apply Saturation'
>
> 'The color in your image will now be reset to its original linear state (though with any gradients and/or light pollution removed if the WIPE module was used). The color in your image may currently appear pale or non-existent. Simply increase saturation to bringing out color. Use the 'Bottom End' and 'Top End' saturation response controls to control how much of the background and stars will be color enhanced respectively. Use the ratio parameters to perfect your white balance'.

Clicking OK allows the COLR module to continue.

The Color module has 9 adjustable parameters and settings

- Histogram -displays the distribution of pixels per channel sorted on intensity. The far left displays the amount of pure dark pixels in the image, the far right displays the amount of pure bright pixels in the image. Note that only pixels that are set in the current active mask are counted.
- [STYLE]- switches the color module's color handling between one of three styles, allowing you flexible control over your color presentation. The [STYLE] parameter is only available when tracking is engaged.
    - [SCIENTIFIC (COLOR CONSTANCY)] uses StarTools' unique ability to display all colors in the scene as if it were evenly illuminated, meaning that even very bright cores of galaxies and nebulas retain the same color throughout, irrespective of their local brightness, Areas of diffuse gas as well as small gas knots in, for example, galaxies all readily stand out and gaseous makeup is easy to compare across the whole image, even in bright cores. The same is true for star temperatures across the image, even in bright, dense star clusters. This mode allows the viewer to objectively compare different parts of the image without suffering from other software's tendency to destroy saturation and shift hue in bright areas. It allows the viewer to

explore the universe in full color, adding another dimension of detail, irrespective of exposure time and subsequent stretching of the data.

- [ARTISTIC, DETAIL AWARE] emulates the way workflows in other software tend to desaturate and distort hue in different parts of the image, making it harder to determine and compare chemical makeup and star temperatures within the image. In such software, brighter areas tend to look washed out and devoid of color although they are not. This mode may be preferable to those who like this look for comparison with older methods of color rendition. ARTISTIC, DETAIL AWARE does use StarTools' tracking information to enhance the behavior of areas whose local brightness have been manipulated (ex. HDR). In these areas StarTools will compensate these manipulations and show more color.
- [ARTISTIC, NOT DETAIL AWARE] is similar to OTHER SOFTWARE, DETAIL AWARE, but does not attempt to use StarTools' tracking information to enhance the behavior of areas whose local brightness have been manipulated (ex. HDR). In these areas StarTools will not compensate these manipulations and color will be incorrectly desaturated and washed out, even though they are not.

- [SATURATION AMOUNT] - specifies the amount of colour saturation as a percentage of the original image. Increasing this value will make colours more vivid. Decreasing this value will make colours less vivid, until the image becomes a grayscale image.
- [BRIGHT SATURATION] -defines the color saturation in the lighter areas. Higher values translate in increased saturation in the very bright areas of the image. Increase this value if, for example, you wish saturation to affect stars.
- [DARK SATURATION] - defines the color saturation in the darker areas. Higher values translate in increased saturation in the very dark areas of the image. Lower this value if, for example, you do not wish the background to respond to changes in saturation.
- [CAP GREEN] - removes any dominant green color information from the image, relying on the fact that not many objects in outer space are green. The COLOR module therefore can be instructed to assume that any dominant green information is the result of noise or unwanted gradients.

 - [TO YELLOW] caps green to yellow, If you find this introduces yellow tints into your image, try TO BROWN.

 - [TO BROWN] caps green to brown.

 NOTE: Exceptions of green areas are those with dominant OIII emissions. Examples may include the Orion Nebula's core (M42) and the Tarantula nebula (NGC 2070).

- [LRGB EMULATION METHOD] allows you to emulate the appearance of many popular methods of color compositing at the touch of a button, even if your data source was not LRGB. The [LRGB EMULATION METHOD] feature is only available when Tracking is engaged.
  - [STRAIGHT CIELAB LUMINANCE RETENTION] manipulates all color psycho-visually optimal in CIELab space, introducing color without affecting brightness.
  - [RGB RATIO, CIELAB LUMINANCE RETENTION] uses a method first proposed by Till Credner of the Max-Planck-Institut and subsequently rediscovered by Paul Kanevsky to use RGB ratios multiplied by luminance in order to better preserve star color. Luminance retention in CIELab color space is applied afterwards.
  - [50/50 LAYERING, CIELAB LUMINANCE RETENTION] uses a method proposed by Robert Gendler, where luminance is layered on top of the color information with a 50% opacity. Luminance retention in CIELab color space is applied afterwards. The inherent loss of 50% in saturation is compensated for your convenience for easier comparison with other methods.
  - [RGB RATIO] uses a method first proposed by Till Credner of the Max-Planck-Institut and subsequently rediscovered by Paul Kanevsky to use RGB ratios multiplied by luminance in order to better preserve star color. No further luminance retention is attempted.
  - [50/50 LAYERING] uses a method proposed by Robert Gendler, where luminance is layered on top of the color information with a 50% opacity. No further luminance retention is attempted. The inherent loss of 50% in saturation is compensated for your convenience for easier comparison with other methods.

- [BIAS SLIDER MODE] - switches the red, green and blue bias sliders to either increase or decrease in bias as the slider is pulled along. Switching between these modes allows you to effectively address any color bias problem you are experiencing in your image in a particular color channel; if you need to reduce a particular channel's prevalence, switch it to 'SLIDERS REDUCE COLOR BIAS'; if you need to increase a channel's prevalence, simply switch the mode to 'SLIDERS INCREASE COLOR BIAS'.

Because channel biases are relative to each other, you could also, for example increase green and blue to achieve a relative bias reduction in red. The BIAS CLIDER MODER parameter allows you to accomplish more intuitive increasing and decreasing of the bias of a particular channel.

To see how an increase in green and blue constitutes a relative decrease in red, set, for example, BIAS SLIDER MODE to SLIDER INCREASE COLOR BIAS, green and blue to 1.5, and leave the red at 1.0. Now switch BIAS SLIDER MODE to SLIDERS REDUCE COLOR BIAS and see how red is automatically set to a reduction of 1.5.

- Red Bias Reduce [RED REDUCE/INCREASE BIAS] increases or decreases (depending on the BIAS SLIDER MODE) a color bias in the red channel. If you image is overly red, increase this parameter in 'reduce' mode. If you image lacks red, increase this parameter in 'increase' mode.
- [GREENBIAS REDUCE] - increases or decreases (depending on the BIAS SLIDER MODE) a color bias in the green channel. If you image is overly green, increase this parameter in 'reduce' mode. If you image lacks green, increase this parameter in 'increase' mode.
- [BLUE BIAS REDUCE] - increases or decreases (depending on the BIAS SLIDER MODE) a color bias in the blue channel. If you image is overly blue, increase this parameter in 'reduce' mode. If you image lacks blue, increase this parameter in 'increase' mode.
- [MASK FUZZ] - specifies the kernel radius of a Gaussian blur to be applied to the current mask. If a mask is used to selectively color the image, using this parameter will ensure smooth (and hence undetectable) transitions between the modified and unmodified parts of the image.

The Color module attempts to keep luminance (perceived brightness) constant and works on the color information only.

Clicking on any pixel in the image will tell Color to use this pixel's color as a white reference to calibrate the entire image to.

The 'Reset' button resets all sliders and settings to their default value.

The Color module demystified

So you had a play with the latest beta and you may have come across the new Color module.
If all is well, the results you've been able to achieve are a lot better, but there are a good number of new features in there.

All of a sudden there are parameters like 'Style' and 'LRGB Method Emulation' and some of you have understandably cried out "I thought StarTools was supposed to make things simpler!".

This post is to demystify what exactly StarTools is doing to your colors and how the Style and LRGB Method Emulation let you wield awesome power, virtually without having to lift a finger.

First up is the 'Style' parameter. Ever since Tracking (and by extent, non-linear processing) was introduced in 1.3, a number of completely new and innovative abilities were bestowed upon StarTools' modules.

We're all familiar with StarTools detail aware noise reduction, deconvolution after stretching, and lately the intelligent 'detail aware' wavelet sharpener. Around 1.3.2 the Color module was revamped with a new way of Color compositing, separating luminance and color processing completely. At the time it seemed like the logical way to implement it, having the Tracking data at our disposal, but it turned out to have some more interesting attributes.

This algorithm, which is now embodied in the 'Scientific (Color Constancy)' setting of the 'Style' parameter, is a new, more scientifically accurate way of representing color in your images. Color perception is a difficult subject and there are no right answers to what constitutes 'the right way' to present color in deep space images. I'd love to elaborate on this, but we'll save that discussion for another time. What does exist, however is a 'worse' way of presenting color in deep space images and it is, unfortunately, surprisingly common. Take this image of the Orion nebula for example;

Illustration 64 – The Orion Nebula

It's not the prettiest image, but there's nothing majorly wrong with that. Or so you'd say... Indeed, this is typical of the best that people using other software come up with in terms of color. But here is the rub;

Illustration 65 - Same as previous image but stretched less

This is the exact same image, same color calibration, just stretched less. The core now appears green (which, incidentally, is correct; OIII emissions are dominant), you might also notice that the stars that are visible are much bluer.

So how come, in the previous image, the core was nearly white?

The answer is that this is the result of (erroneously) stretching the color data along with the luminance (brightness) as well, which is strangely very much common practice, even in other software that prides itself on scientific accuracy. Color is a product of discrepancies in the red, green and blue channels. Stretching these channels non-linearly deforms, stretches, squashes and skew these discrepancies based on their in-channel brightness. The result is that hues and saturation levels vary wildly as the user starts stretching the data in order to recover more luminance detail.

Of course, the notion that things out there in space 'magically' change their color based on how we stretch our image would be a fairly ludicrous proposition. Yet, sadly, here we are, with a majority of images out there suggesting that this is the case.

Now look at the two images again. The core is clearly green in the second (less stretched) image, while the core in the first image is so pale that we might have missed it is actually green (indeed many astrophotographers throw their hands up in the air and just depict it as white). However, notice the area adjacent to the core at 3 o'clock (in the second image it is completely absent). It is decidedly more green. Could the two areas be of similar chemical make up?

In fact they are! It is just neigh impossible to see, as stretching the luminosity component has (for no good reason at all) drained the core of its color, compared to the area adjacent to it.

And here is where StarTools' Tracking aided color compositing algorithm comes in;

Illustration 66 – The new image. See text.

New let's ignore for a minute that the colors might appear a little oversaturated. In the interest of fairness I applied the same settings for illustration 64 and illustration 66. I had to be a good bit more aggressive with the saturation to show any color in illustration 64.

What we're seeing here is 'color constancy'. No matter how the image was stretched, the colors are absolutely comparable between areas that vary wildly in their dynamic range. The area adjacent to the core at 3 o'clock is the exact same green as the core. Also spare a moment to look at the stars. They now show color, no matter how dull or bright they are. The full visible black body emission spectrum is covered and what's most important - temperatures are completely stable and independent of how dull or bright they were recorded and subsequently stretched. This is because StarTools stays true to the color data as initially recorded, undoing all the stretching (thanks to Tracking!) to the luminance data to arrive at the 'virgin' colors as they were recorded.

This is how color in space should be presented from a scientific point of view - reproducibility no matter the exposure time or who processed it, with true separation from the way luminance was processed. Faint Andromedas should produce the same colors as bright Andromedas. Short exposure star fields should produce the same colors as long exposure starfields, etc.

Here is another striking example;

Illustration 67 - Before

vs

Illustration 68 - After

Notice how star colors are recovered until well into the core of the globular cluster, while the viewer is able to objectively compare star temperatures all the way; colors are constant and comparable throughout the DSO.

Now, as of 1.3.5 there is actually a way to 'dumb down' (which perversely was a lot of work to implement and get right!) color compositing to the levels of other software. StarTools is all about enabling the user, not about pushing philosophies, so if you want to create images with the 'handicapped' traditional look you now can, using the 'Artistic' setting for the 'Style' parameter.

As a StarTools user you may have felt that your colors were never quite like 'the other guys'; more vivid, just more colorful. And you would have been right. You may have secretly liked it but you may have been uneasy being proud of the result as it is 'different' to what a lot of other folks have been producing (your Andromeda may have had a 'strange' yellow core, your M42 may have had an 'odd' green core, your M16 may have had many more tints of red and pink, or your stars seemed almost too CGI-pretty to be true). Now you know why, and now you know you can be just as proud at your images - perhaps even more, as they have been more scientifically correct/valuable to look at than the other guy's.

Now, on to the enigmatic 'LRGB Method Emulation' setting.
Over the past two decades, astrophotographer have struggled with compositing color (e.g. combining separate luminance and color information) images, leading to a number of different LRGB compositing methods. I won't go into them here, but all attempt to invariably 'marry' luminance and color information. The different techniques all produced different hues and saturations for the same data. And what is true for all of them is that they involved a lot of layering and pixel math to produce.

Because StarTools separates color from luminance until the very last moment (that moment being when you use the Color module), really any method of color compositing can be chosen for combining the data at that moment. The 'LRGB Method Emulation' simply lets you choose from a number of popular compositing techniques at the click of a button. You will notice that all these different settings have a subtle (or sometimes not-so-subtle) impact on the hues (and sometimes saturation and luminance) that are produced. It's just that easy.

Even if your data was from a non-LRGB source, it still works as StarTools will have still separated color from luminance during processing by creating a synthetic luminance frame from your RGB-only data where needed. Conversely, as you may have seen in the latest YouTube video, processing LRGB data is now no different than processing DSLR or OSC data. Luminance is separated anyway! The only difference is that you load your data through the LRGB module. It's a big step towards a unified yet extremely powerful workflow for all data sources.

Crop

Overview

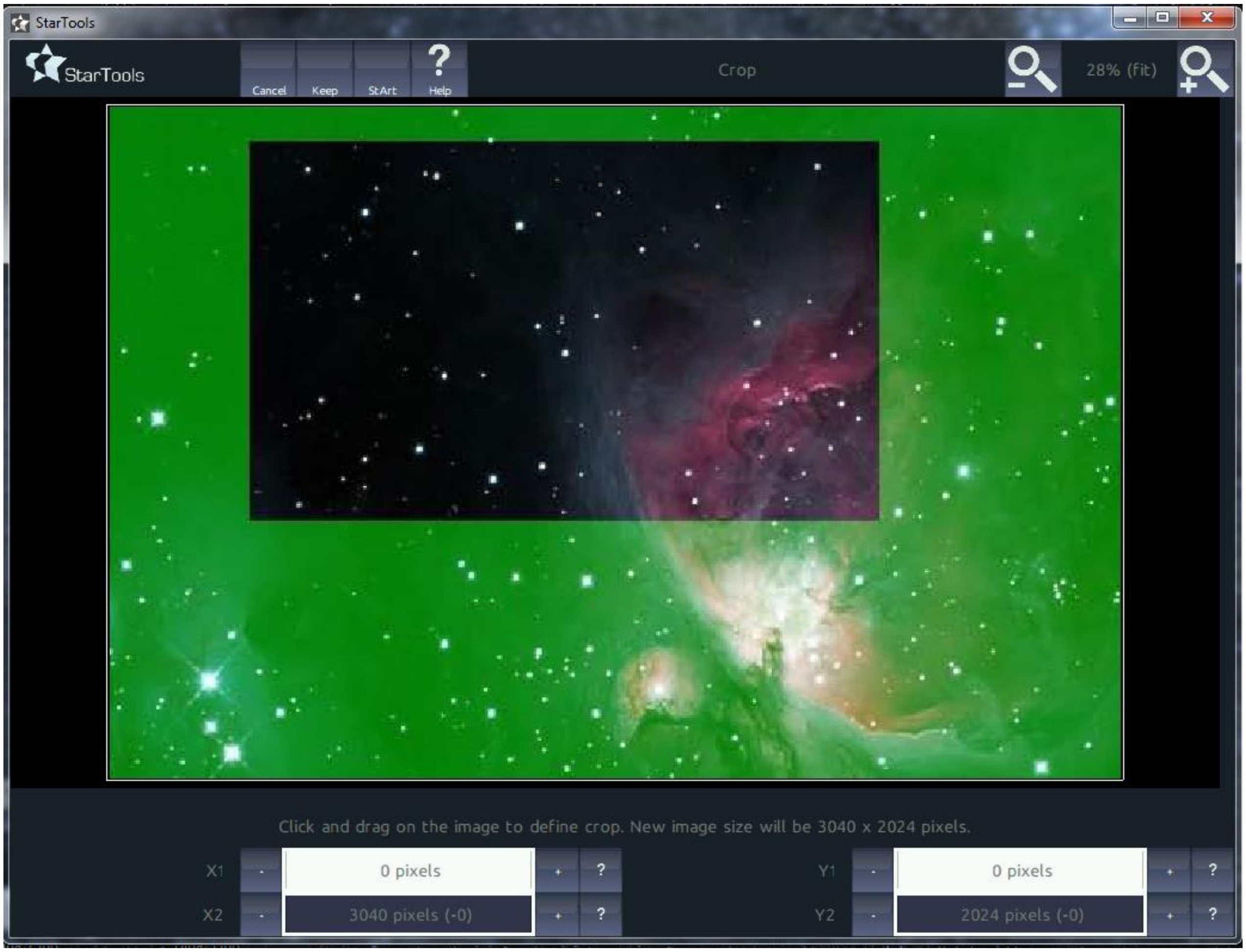


Illustration 69 - The CROP module interface

The Crop module allows the user to crop the image. This is particularly useful for framing a subject, or to get rid of any stacking artefacts that are commonly found close to the edges.

Usage

CROP is a simple cropping tool. The preset StArt button will automatically attempt to detect and remove stacking artifacts. Crop has just 4 parameters:

- X1 -specifies the first X coordinate.

- Y1 -specifies the first Y coordinate.
- X2 -specifies the second X coordinate.
- Y2 -specifies the second Y coordinate.

The Crop module will show which parts of the image will be cut off by showing those pixels coloured in green. Any pixels in the image that will be kept, will be shown in their original pixels. Clicking the 'Crop' button will finalise the cropping and return the user to the main screen.

Mirror H

Overview

[MIRROR H] flips the image horizontally. It has no settings screen.

Mirror V

Overview

[MIRROR V] flips the image vertically. It has no settings screen.

ROTATE

Overview

The ROTATE module simply rotates your image.

Usage

- Rotate provides 3 preset angle buttons: 90, 180 and 270 degrees.
- [Angle] defines the angle by which the image should be rotated.

Masks

What is a Mask

The Mask feature is an integral part of StarTools. Many modules use a mask to operate on specific pixels and parts of the image, leaving other parts intact. Besides operating only on certain parts of the image, it allows the many modules in StarTools to perform much more sophisticated operations.

You may have noticed that when you launch a module that is able to apply a mask, the pixels that are set in the mask will flash three times in green. This is to remind you which parts of the image will be affected by the module and which are not. If you just loaded an image, all pixels in the whole image will be set in the mask, so every pixel will be processed by default. In this case, when you launch a module that is able to apply a mask, the whole image will flash in green three times.

Green coloured pixels in the mask are considered 'on'. That is to say, they will be altered/used by whatever processing is carried out by the module you chose. 'Off' pixels (shown in their original colour) will not be altered or used by the active module. Again, please note that, by default all pixels in the whole image are marked 'on' (they will all appear green).

For example, an 'on' pixel (green coloured) in the Sharp module will be sharpened, in the Wipe module it will be sampled for gradient modelling, in Synth it will be scanned for being part of a star, in Heal in will be removed and healed, in Layer it will be layered on top of the background image, etc.

To recap;

> If a pixel in mask is 'on' (coloured green), then this pixel is fed to the module for processing.
>
> If a pixel in mask is 'off' (shown in original color), then tell the module to 'keep the pixel as-is, hands off, do not touch or consider'.

The Mask Editor

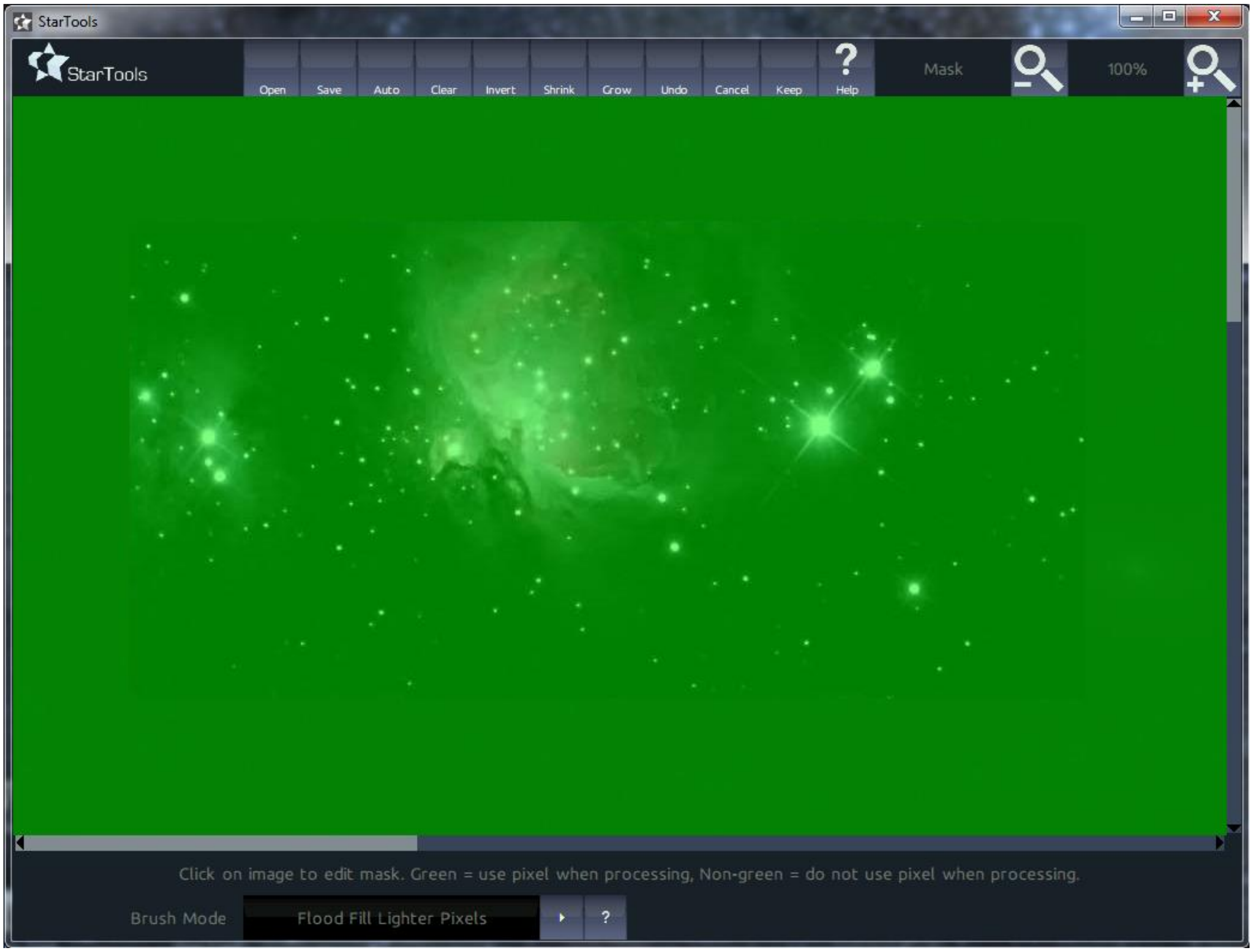


Illustration 70 - *The Mask Editor showing an image with all pixels selected for processing.*

The Mask Editor is accessible from the main screen, as well as from the different modules that are able to apply a mask. The button to launch the Mask Editor is labelled 'Mask'. When launching the Mask Editor from a module, pressing the 'Keep' or 'Cancel' buttons will return StarTools to the module you pressed the 'Mask' button in.

As indicated by the 'Click on the image to edit mask' message, clicking on the image will allow you create or modify a Mask. What happens when you click the image depends on the selected 'Brush Mode'. While some of the 'Brush Modes' (see below) seem complex in their workings, they are quite intuitive to use.

There are three 'Brush Modes'

- Flood Fill Lighter Pixels
    - Clicking a non-green pixel will, starting from the clicked pixel, recursively fill the image with green pixels until it finds that;
        - All neighbouring pixels of a particular pixel are already filled (green).
        - Or the pixel under evaluation is darker than the original pixel clicked. Clicking on a green pixel will, starting from the clicked pixel, recursively turn off any green pixels until it can no longer find any green neighbouring pixels.
- Flood Darker Pixels;
    - Clicking a non-green pixel will, starting from the clicked pixel, recursively fill the image with green pixels until it finds that;
        - All neighbouring pixels of a particular pixel are already filled (green).
        - Or the pixel under evaluation is lighter than the original pixel clicked.
    - Clicking on a green pixel will, starting from the clicked pixel, recursively turn off any green pixels until it can no longer find any green neighbouring pixels.
- Single Pixel;
    - Clicking a non-green pixel will make a non-green pixel turn green.
    - Clicking a green pixel will make green pixel turn non-green.

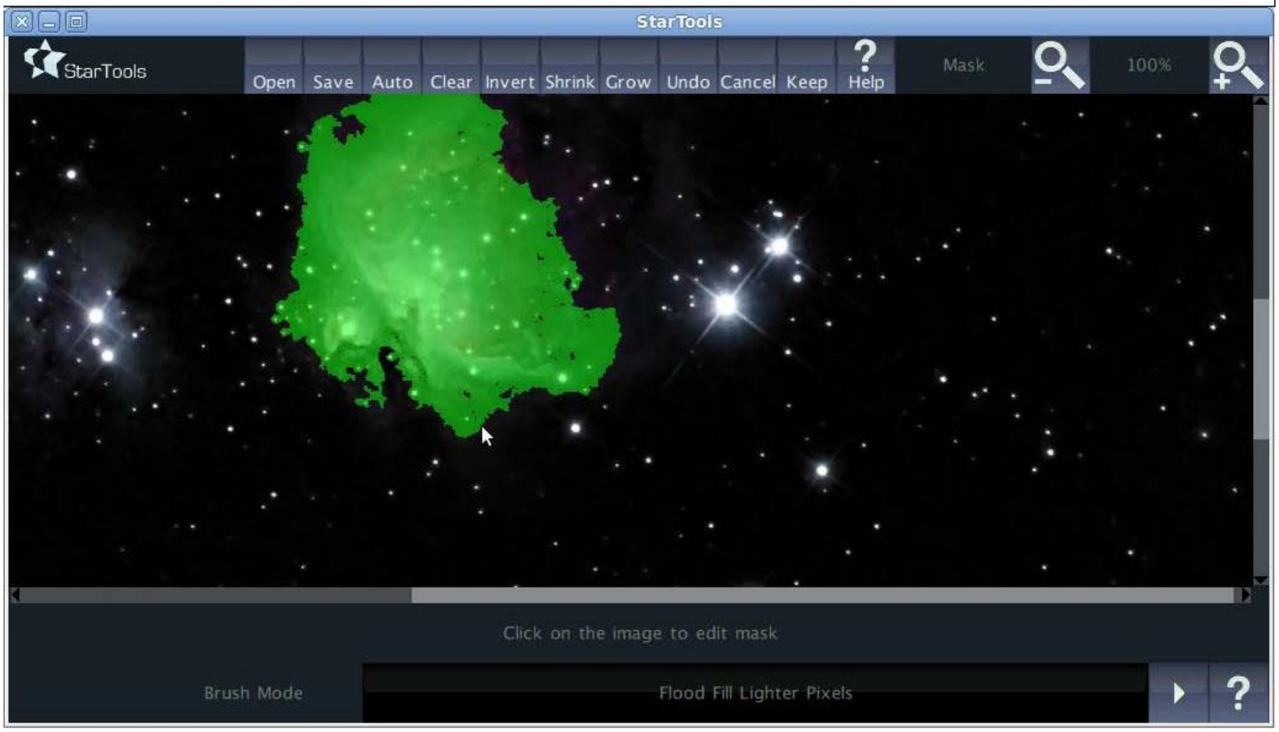


Illustration 71 - Clicking next to a feature (note the position of the mouse cursor) that is brighter than its surroundings will select it in a single click using the 'Flood Fill Lighter Pixels' 'Brush Mode'.

These different 'Brush Modes' help in quickly selecting (and de-selecting) features in the image. For example, while in 'Flood Fill Lighter Pixels' mode, try clicking next to a bright star or feature to select it (see Illustration 71). Click anywhere on a clump of 'on' (green) pixels, to toggle the whole clump off again.

Please note that, depending on the size of the image, the mode selected and the amount of pixels that need to be flood filled, there may be a minor delay in response between clicking on the image and seeing a result. This is normal behaviour.

Illustration 72 - Pressing the 'Invert' button on the mask in Illustration 53.

Apart from the different brush modes, various other functions exist to make editing and creating a Mask even easier;

- The 'Save' button allows you to save the current mask to a standard TIFF file.
- The 'Open' button allows you to import a Mask that was previously saved by using the 'Save' button. Note that the image that is being opened to become the new Mask, needs to have the same dimensions as the image the Mask is intended for.
- The 'Auto' button is a very powerful feature that allows you to automatically isolate features. For more details, please see the sub chapter on the Auto feature on page 156.
- The 'Clear' button turns off all green pixels (i.e. it deselects all pixels in the image).
- The 'Invert' button turns on all pixels that are off, and turns off all pixels that were on[11].

[11]Tip! To quickly turn on all pixels, click the ‘clear’ button, then the ‘invert’ button.

- The 'Shrink' button turns off all the green pixels that have a non-green neighbour, effectively 'shrinking' any selected regions.
- The 'Grow' button turns on any non-green pixel that has a green neighbour, effectively 'growing' any selected regions.
- The 'Undo' button allows you to undo the last operation that was performed.

Tip: to quickly turn on all pixels, click the 'Clear' button, then the 'Invert' button.

As with the different modules in StarTools, the 'Keep' and 'Cancel' buttons work as expected; 'Keep' will keep the edited Mask and return, while 'Cancel' will revert to the Mask as it was before it was edited and return.

As mentioned before, when launching the Mask Editor from a module, pressing the 'Keep' or 'Cancel' buttons will return StarTools to the module you pressed the 'Mask' button in.

The Auto Feature

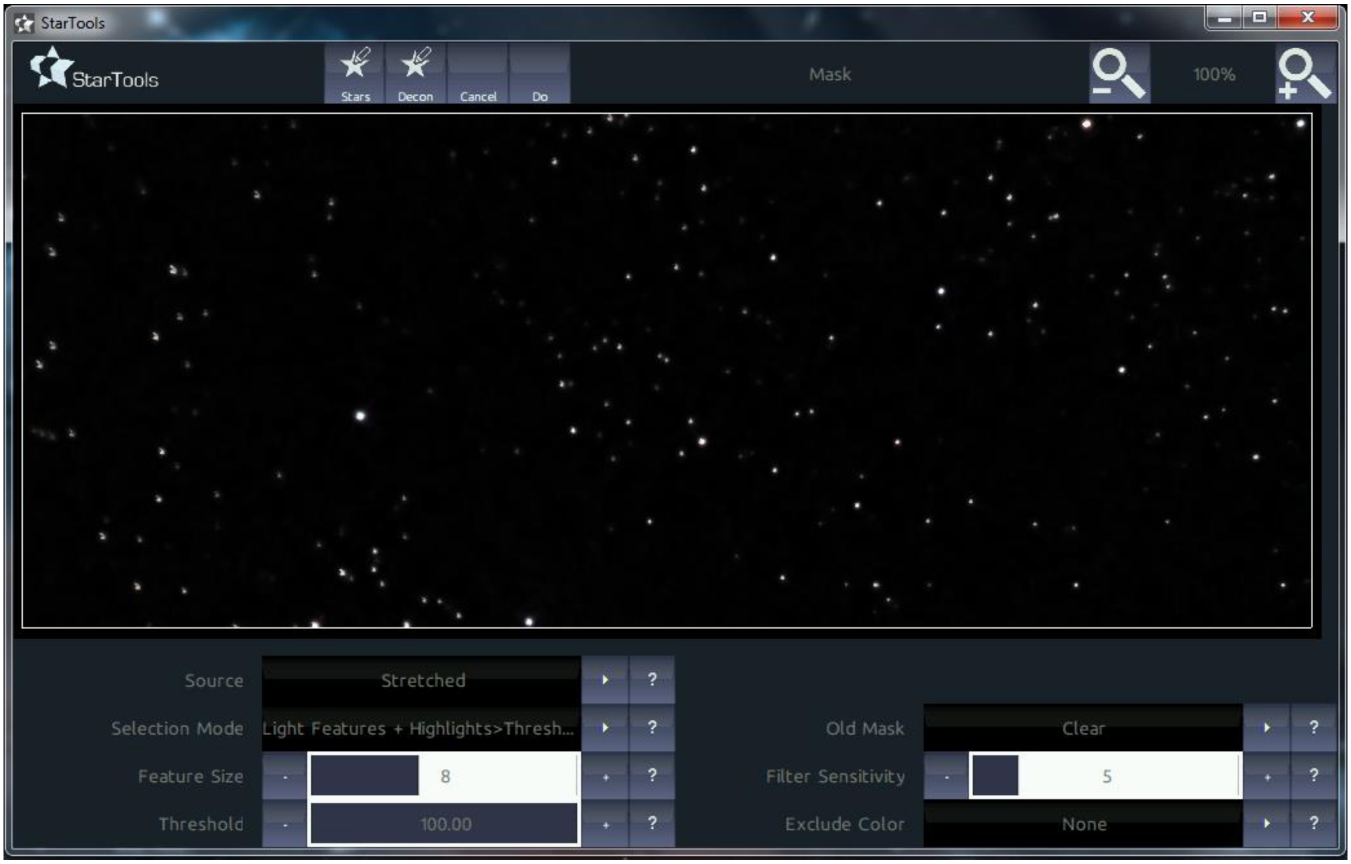


Illustration 73 - Mask's 'Auto' feature interface.

The 'Auto' function is a very powerful feature for quickly isolating features of interest (stars, noise, hot or dead pixels, etc.). For example, isolating only the stars is in an image is a necessity for obtaining any useful results from the 'Magic' module (see page 91).

For examples on how the Auto feature is used in conjunction with the different modules in StarTools, please refer to the star detection, noise reduction and hot pixel removal examples on page 158, 161 and 168 respectively.

The type of features to be isolated are controlled by the 'Selection Mode' parameter

- Light Features + Highlight>Threshold - is a combination of two selection algorithms. One is the simpler 'Highlight>Threshold' mode, which selects any pixel whose brightness is brighter than a certain percentage of the maximum value (see the 'Threshold' parameter below). The other selection algorithm is 'Light Features' which selects high frequency components in an image (such as stars, gas knots and nebula edges), up to a certain size (see 'Max Feature Size' below) and depending on a certain sensitivity (see 'Filter Sensitivity' below'). This mode is particularly effective for selecting stars. Note that if the 'Threshold' parameter is kept at 100%, this mode produces results that are identical to the 'Light Features' mode.

- Light Features -selects high frequency components in an image (such as stars, gas knots and nebula edges), up to a certain size (see 'Max Feature Size' below) and depending on a certain sensitivity (see 'Filter Sensitivity' below').
- Highlight>Threshold - selects any pixel whose brightness is brighter than a certain percentage of the maximum value (see the Threshold parameter below). If you find this mode does not select bright stars with white cores that well, open the 'Levels' module and set the 'Normalization' a few pixels higher. This should make light features marginally brighter and dark features marginally darker.
- Dead Pixels<Threshold - selects dark high frequency components in an image (such star edges, halos introduced by over sharpening, nebula edges and dead pixels), up to a certain size (see 'Max Feature Size' below) and depending on a certain sensitivity (see 'Filter Sensitivity' below') and whose brightness is darker than a certain percentage of the maximum value (see the Threshold parameter below). It then further narrows down the selection by looking at which pixels are likely the result of CCD defects (dead pixels).
- Hot Pixels>Threshold - selects high frequency components in an image up to a certain size (see 'Max Feature Size' below) and depending on a certain sensitivity (see 'Filter Sensitivity' below'). It then further narrows down the selection by looking at which pixels are likely the result of CCD defects or cosmic rays (also known as 'hot' pixels). The 'Threshold' parameter controls how bright hot pixels need to be before they are potentially tagged as 'hot'. Note that a 'Threshold' of less than 100% needs to be specified for this mode to have any effect. Noise Fine - selects all pixels that are likely affected by significant amounts of noise. Please note that other parameters such as the 'Threshold', 'Max Feature Size', 'Filter Sensitivity' and 'Exclude Color' have no effect in this mode.

- Noise Coarse -selects all pixels that are likely affected by significant amounts of noise. This algorithm is more aggressive in its noise detection and tagging than 'Noise Fine'. Please note that other parameters such as the 'Threshold', 'Max Feature Size', 'Filter Sensitivity' and 'Exclude Color' have no effect in this mode.

- Sobel (Edge) – selects all pixels that are likely to belong to the edge of a feature. Use the 'Threshold' parameter to set sensitivity where lower values make the edge detector more sensitive.

As has been alluded to above, some of the selection algorithms are controlled by additional parameters

- Exclude Color - tells the selection algorithms to not evaluate specific colour channels when looking for features. This is particularly useful if you have a predominantly red, purple and blue nebula with white stars in the foreground and, say, you'd want to select only the stars. By selecting 'Exclude Color' 'Purple (Red + Blue), you are able to tell the selection algorithms to leave features in the nebula alone (since these features are most prominent in the red and blue channels). This greatly reduces the amount of false positives.

- Max Feature Size - specifies the largest size of any feature the algorithm should expect. If you find that stars are not correctly detected and only their outlines show up, you may want to increase this value. Conversely, if you find that large features are being inappropriately tagged and your stars are small (for example in wide field images), you may reduce this value to reduce false positives.

- Filter Sensitivity - specifies how sensitive the selection algorithms should be to local brightness variations. A lower value signifies a more aggressive setting, leading to more features and pixels being tagged.

- Threshold - specifies a percentage of full brightness (i.e. pure white) below, or above which a selection algorithm should detect features.

Example 1-Star Detection

Star detection has been simplified in this version of StarTools. The following example demonstrates how the Auto feature can be used for automated star detection.

We open up an image of a star field and go into the Mask editor, as can be seen in Illustration 74. Note that we cleared the mask in this illustration for clarity, but clearing the mask is not a necessary step.

Illustration 74 – An image of a star field.

From here, we click the 'Auto' button, after which a screen will appear similar to that of Illustration 75.

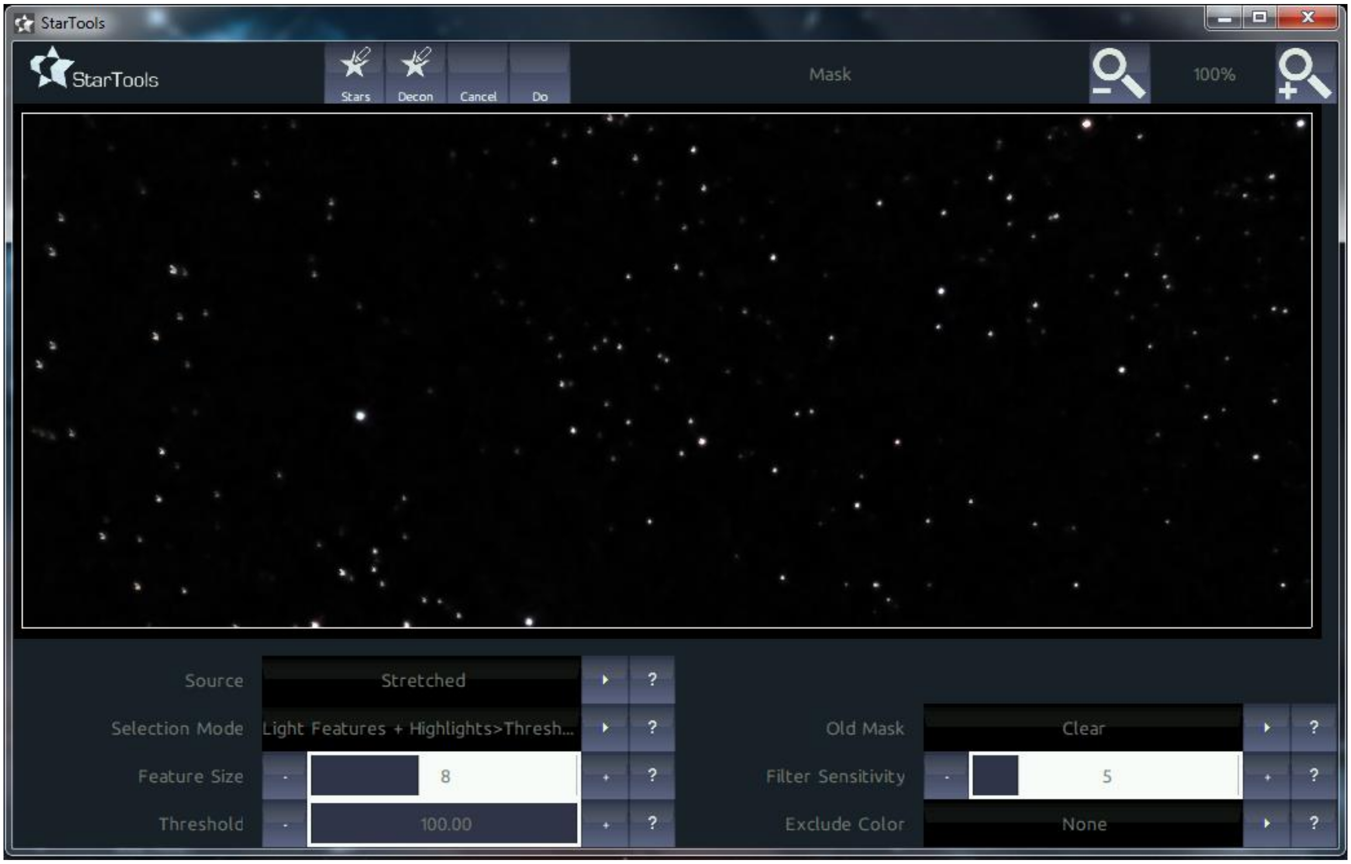


Illustration 75 - **The 'Auto' feature screen**

Click on the ‘Stars’ button and start the detection process by clicking the 'Do' button.

Illustration 76 - **The mask with only stars selected.**

After a short time, the 'Auto' function will return you to the Mask editor screen, with the resulting new selection set (see Illustration 76). Here you can evaluate the result and touch up the mask if necessary.

Example 2: Noise Reduction

The following example demonstrates how the Auto feature can be used for Noise reduction. Please note that is just one of several ways of reducing noise in StarTools. To find out about other types of noise reduction in StarTools, please see the Denoise and Color module chapters on pages122 and 133 respectively.

We open up an image of the Keyhole Nebula and go into the Mask editor, as can be seen in Illustration 65. Note that we cleared the mask in this illustration for clarity - clearing the mask is not a necessary step.

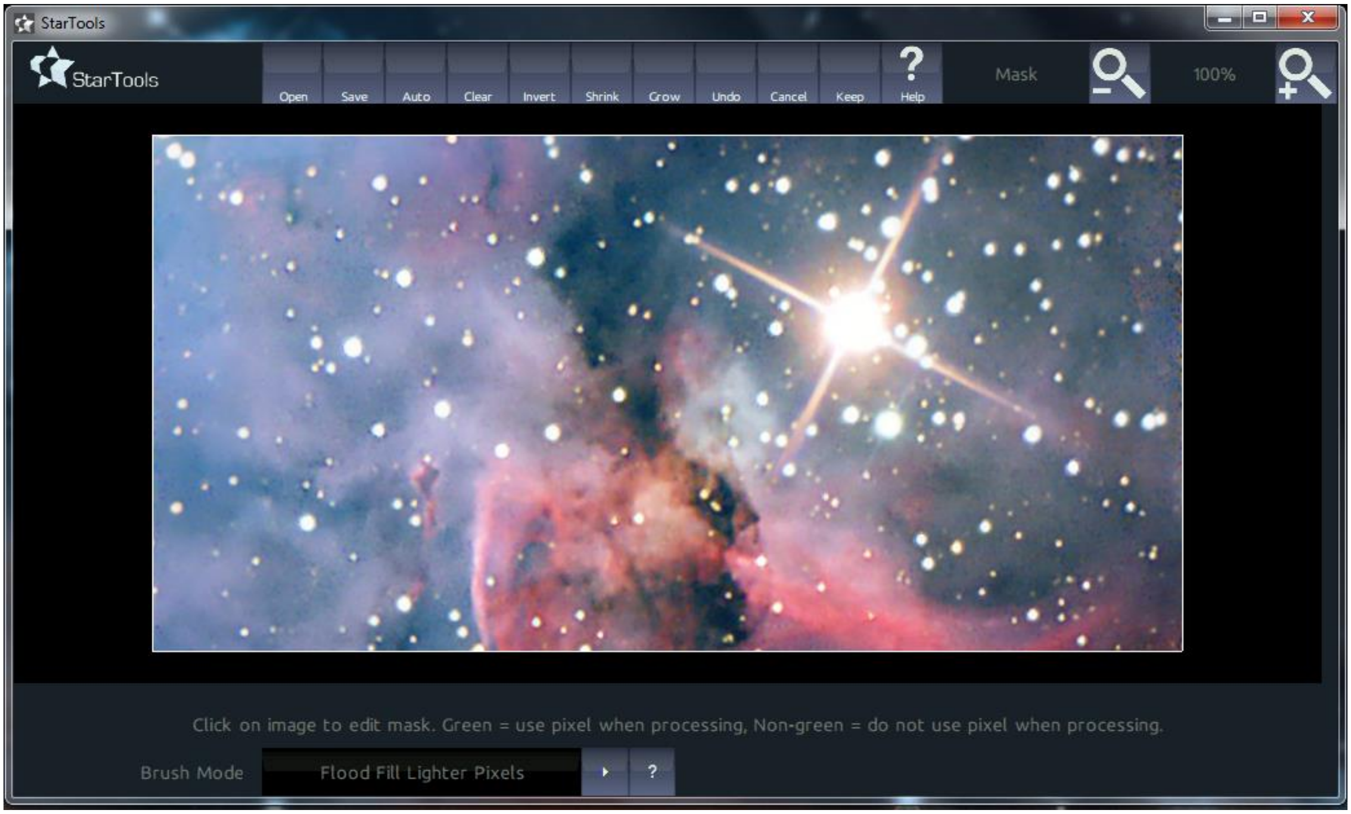


Illustration 77 - *An image of the Keyhole Nebula (image courtesy of Rolf Wahl Olsen).*

Illustration 78 - The 'Auto' feature screen.

Having clicked the 'Auto' button, we select 'Noise' as the 'Selection Mode' - refer Illustration 78. Use the Feature Size and Threshold to define noise grain and sensitivity. We can now click 'Do' to commence the Mask generation.

Illustration 79 - The newly generated mask with 'noisy' pixels tagged.

After a short amount of time, the 'Auto' function will return you to the Mask editor screen, with the resulting new selection set (see Illustration 79).

As seen in Illustration 79, the selection algorithm has tagged 'noisy' pixels throughout the image. We are now ready to use this selection of pixels in another module, the Layer module.

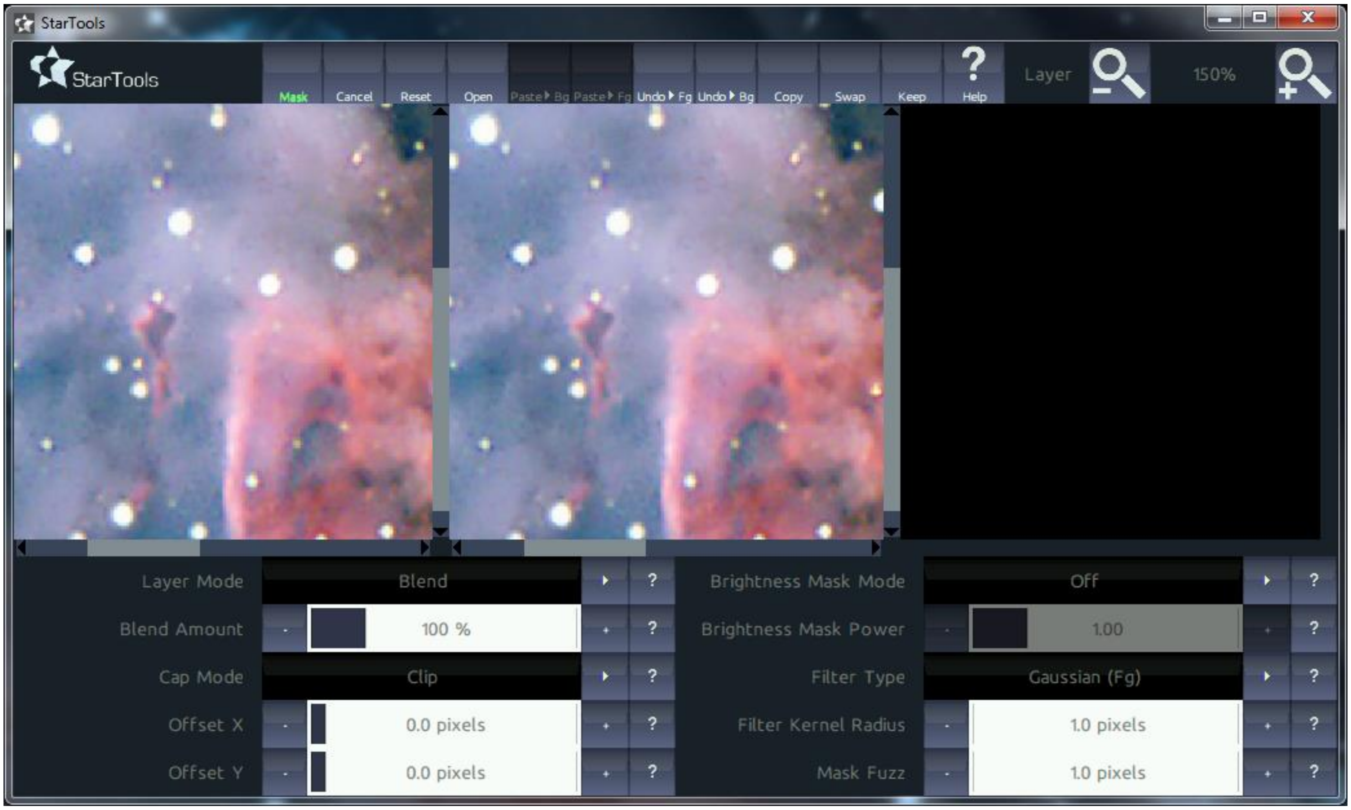


Illustration 80 - The Layer module.

We now launch the Layer module (see Illustration 80). The Layer module (see page 105) allows us to layer one image on top of the other in a manner specified by its parameters. We will use this module to layer a blurred version of the original image on top of the original image, but only for those pixels in our 'noise' mask that are set, leaving the 'good' pixels intact and unblurred.

The center Foreground Layer will blink the noise mask three times in rapid succession.

Illustration 81 - The Layer module with cloned image and layered result.

The right image, which is the result of the layering operation will not show improvement yet (see Illustration 81).

Select ‘Gaussian (Fg)’ as the ‘Filter Type’ and adjust the ‘Filter Kernel Radius’ to blur the noise.

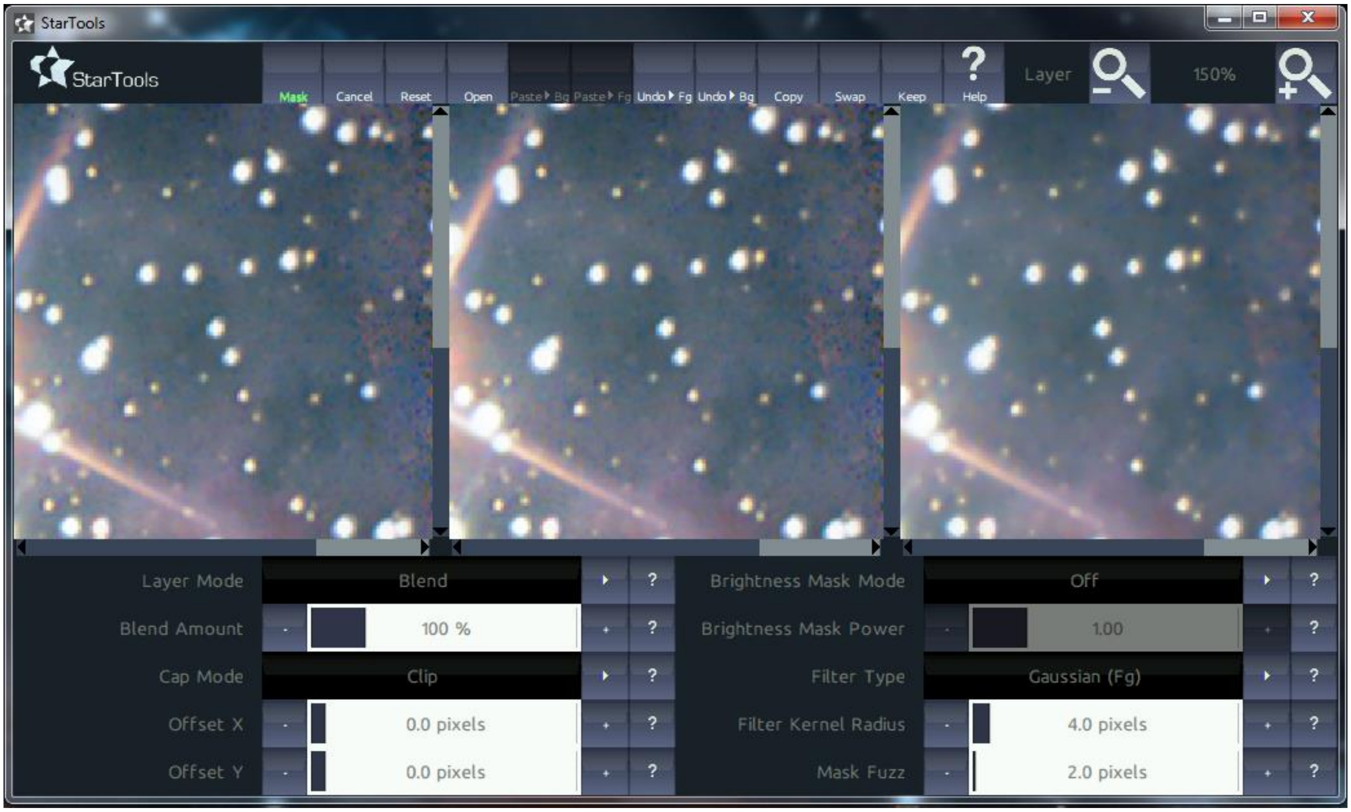


Illustration 82 - The Layer module with cloned image and result with Gaussian blur and Mask Fuzz applied.

It is only after we apply a Gaussian Blur and some Mask Fuzz that we can see a marked improvement in the right image (see Illustration 82); noise has been reduced significantly but detail has not suffered.

Example 3: Hot Pixel Removal

The following example demonstrates how the Auto feature can be used to remove hot pixels.

NOTE: This procedure uses the Heal module which is not available when tracking is active.

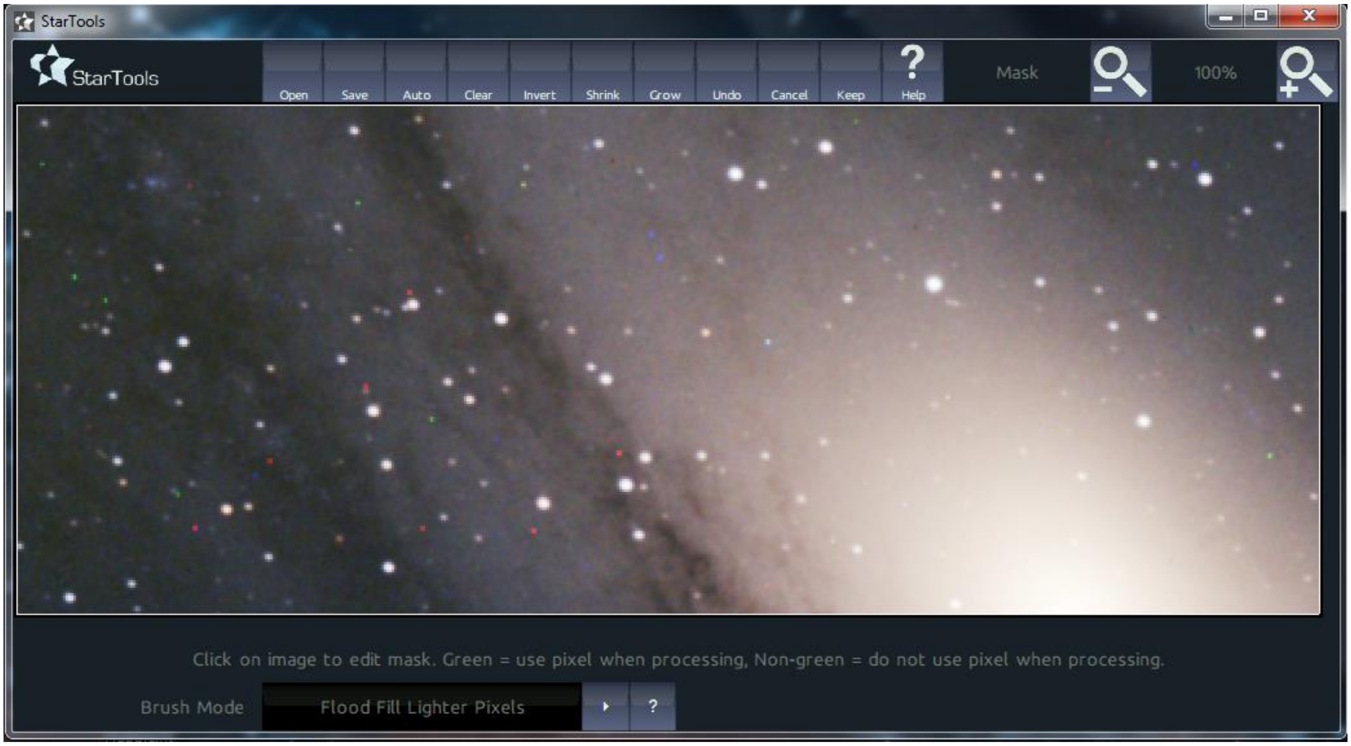


Illustration 83 - M31 in the Mask editor.

We open up an image of M31 which has quite a few hot pixels (some red dots, as well as some green and blue dots are clearly visible in Illustration 83). We then go into the Mask editor. Note that we cleared the mask in Illustration 83 for clarity, but clearing the mask is not a necessary step.

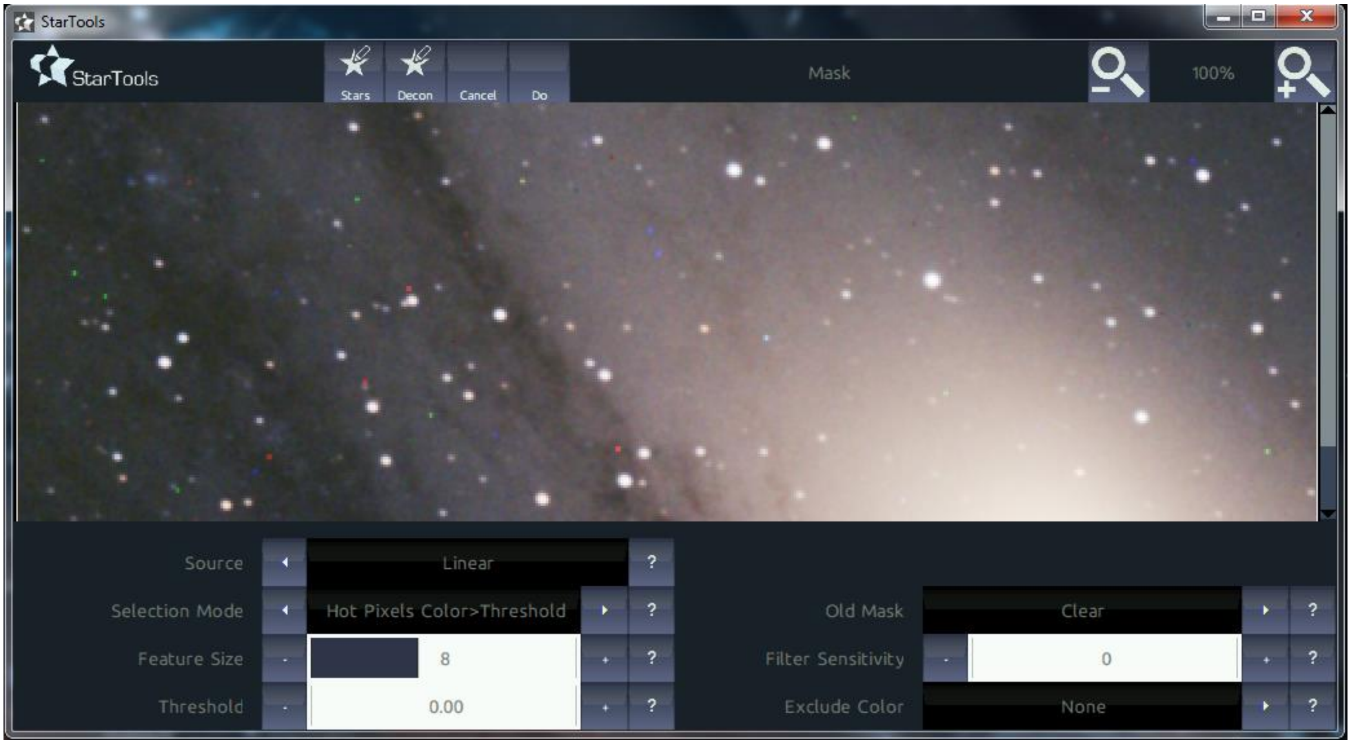


Illustration 84 - Setting up the 'Auto' function for hot pixel removal.

We then proceed by clicking the ‘Auto’ button and setting up the parameters for hot pixel removal. We select 'Hot Pixels Mono>Threshold' or 'Hot Pixels Color>Threshold' as our 'Selection Mode' depending on the image type as well as the most aggressive values possible for hot pixel removal.

We set 'Threshold' to 0%, meaning we don't care how bright a hot pixel must be before we remove it – just remove anything we encounter.

We set 'Filter Sensitivity' to 0, meaning we want to identify as many features as possible for subsequent analysis, no matter how faint. This allows us to detect 'warm' pixels. These are pixels that are not completely defunct but still yield a higher value than they should.

Once you are satisfied with the parameters, click the 'Do' button to commence the Mask generation.

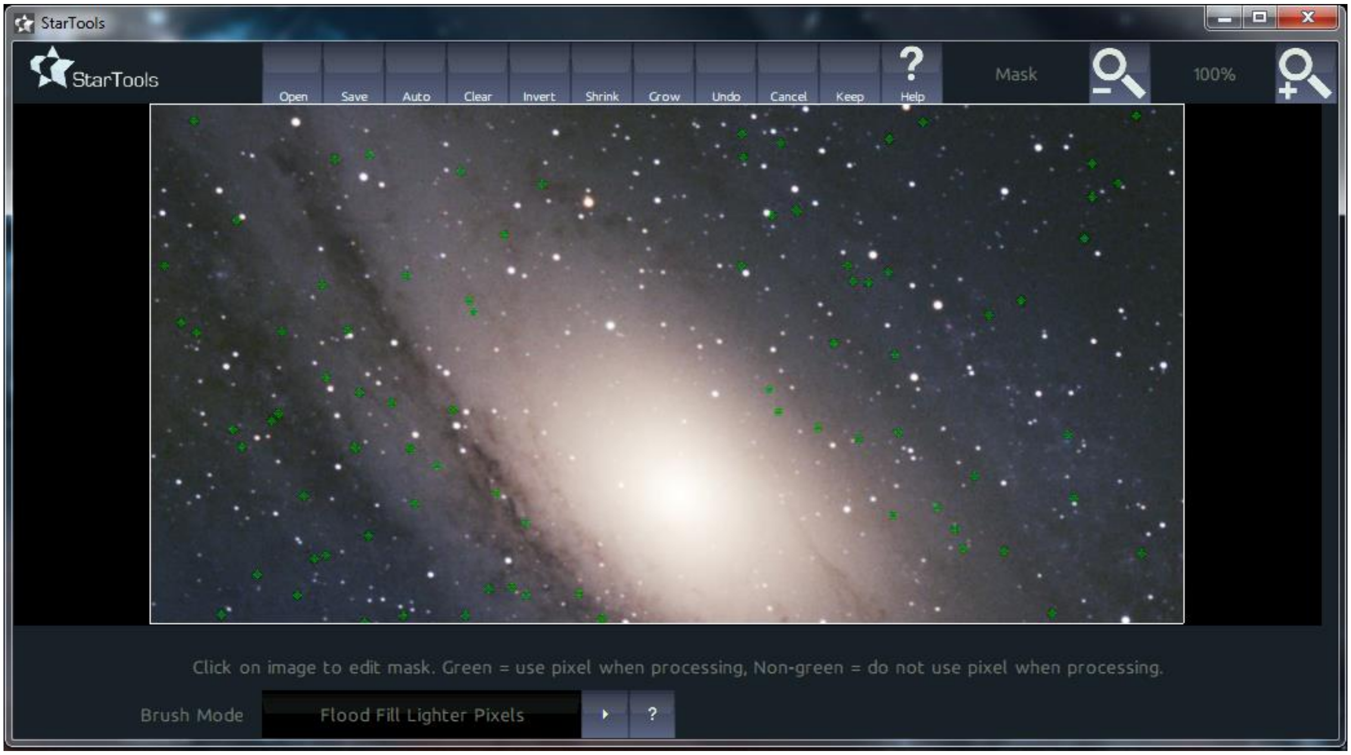


Illustration 85 - The mask with hot pixels after several clicks on the 'Grow' button.

After a short time, the 'Auto' function will return you to the Mask editor screen with all hot pixels tagged. However, unless zoomed in, the selection may be hard to see, as hot pixels are typically one pixel in size.

While the hot pixels should have been detected successfully, their ill effects may stretch beyond just a single pixel. This is typically true for images that have been debayered such as images taken with an OSC, DSLR, compact camera or webcam. It is therefore prudent to grow the selection by a couple of pixels. To do this, simply click on the 'Grow' button a couple of times. Your mask will now look similar to the one in Illustration 85 - notice how the hot and warm pixels are now clearly visible as green dots.

All we have to do now, is remove the aberrant pixels. We do this using the Heal module (see also page 83).

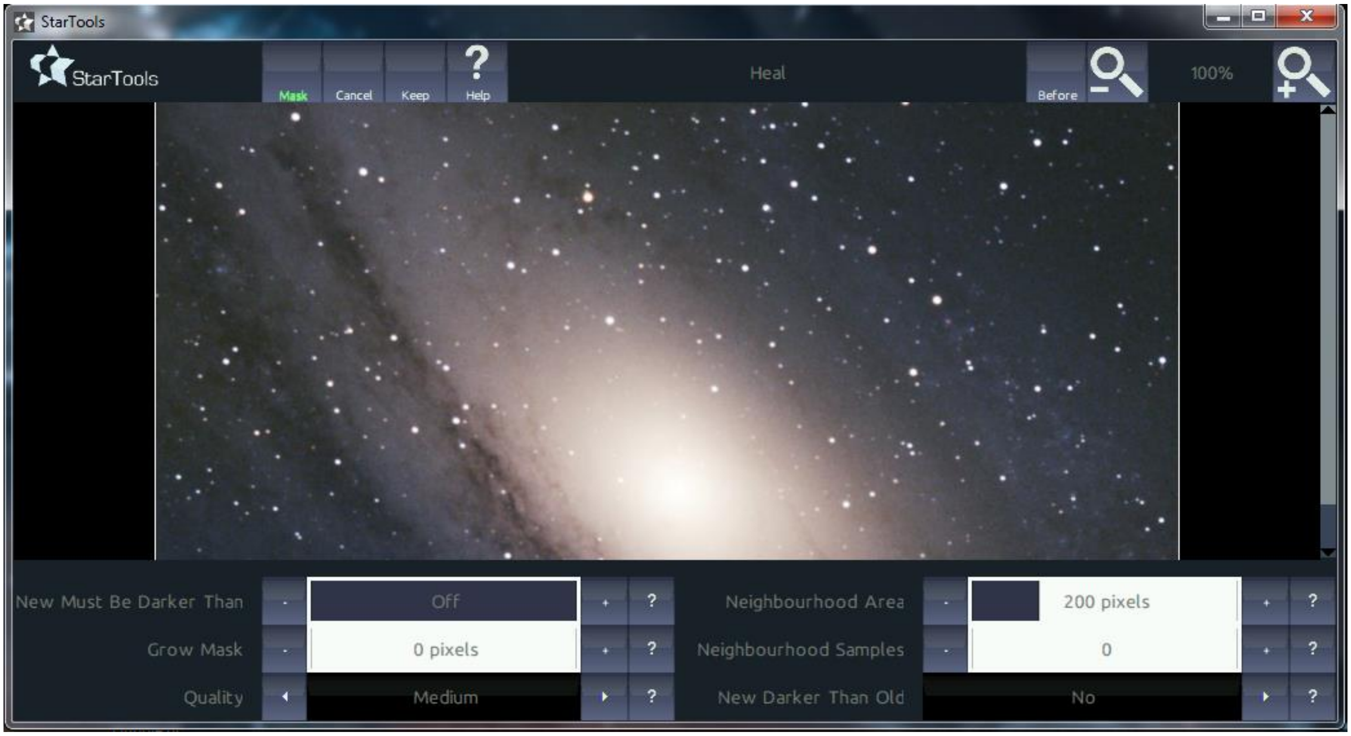


Illustration 86 - The Heal module showing a healed, hot pixel free version of M31 using the hot pixel mask.

Simply launching the Heal module will yield a result similar to that of Illustration 86. The hot and warm pixels will have disappeared. Note that if we forgot to use the 'Grow' button in the Mask editor, we can use the 'Grow Mask' slider to do it here on a temporary basis - that is, without modifying the active Mask.

Acknowledgements

StarTools and this manual were made possible by the contributions of the following people;

Stephen Statts, USA

Alan Mehan, Australia

Charles Kuehne, USA

Jim Misti, USA

Marc Aragnou, Australia

Pete Scully, South Africa

Roger Groom, Australia

Rolf Wahl Olsen, New Zealand

Rowland Cheshire, Australia

Warren Maag, Australia

The author wishes to thank them for providing images, equipment and their valuable time beta testing StarTools.

Appendix A

A Suggested StarTools Workflow

- Load Data (use Open or LRGB)
    - Indicate data is still linear to switch on tracking
- Use Develop or AutoDevelop to stretch image
- Fix stacking artifacts with Crop
- Fractionally Bin the data if 1 unit of data does not fit in 1 pixel (reduces noise)
- Wipe away light pollution, gradients, vignetting and amp glow with Wipe module
- Redo Develop or AutoDevelop for final global stretch
- Use Deconvolution
- Use Contrast for medium-large local dynamic range optimization
- Use HDR for medium-small local dynamic range optimization
- Use Sharp (with mask from Deconvolution) for detail enhancement
- Color calibration with Color module
- Switch Tracking off for final noise reduction