StarTools

# Module features and documentation

StarTools comprises several modules with deep, state-of-the-art functionality that rival (and often improve on) other software packages.

# Introduction

Do not be fooled by StarTools' simple interface. You are forgiven if, at first glance, you get the impression StarTools offers only the basics. Nothing could be further from the truth!

StarTools goes deep - very deep in fact. It is just not "in your face" about it, and you can still get great results without delving into the depths of its capabilities. It is up to you how you wish to approach image processing.

If you are a seasoned photographer looking to get more out of your data, StarTools will allow you to visibly gain the edge with novel, brute-force techniques and data mining routines that have only just become viable on modern 64-bit multi-core CPUs, GPU compute power, and increases in RAM and storage space.

If you are a beginner, StarTools will assist you by making it easy to achieve great results out-of-the box, while you get to know the exciting field of astrophotography better.

Whatever your situation, skills, equipment and prior experience, you will find that working with StarTools is quite a bit different than any software

you may worked with prior. And in astrophotography, that tends to be a *good* thing!

# Quick Start Tutorial: a quick generic work flow

Getting to grips with new software can be daunting, but StarTools was designed to make this as painless as possible. This quick, generic work flow will get you started.

While processing your first images with StarTools, it may help knowing that the icons in the top two panels roughly follow a recommended workflow when read top to bottom, left to right.

The screenshots in this quick start tutorial, use an intentionally modest, flawed DSLR dataset to demonstrate some common pitfalls. If, however, you process high quality OSC, mono CCD, space telescope or space probe datasets, whether they be narrowband or visual spectrum datasets, you will be happy to know that the general workflow and considerations are substantially the same.

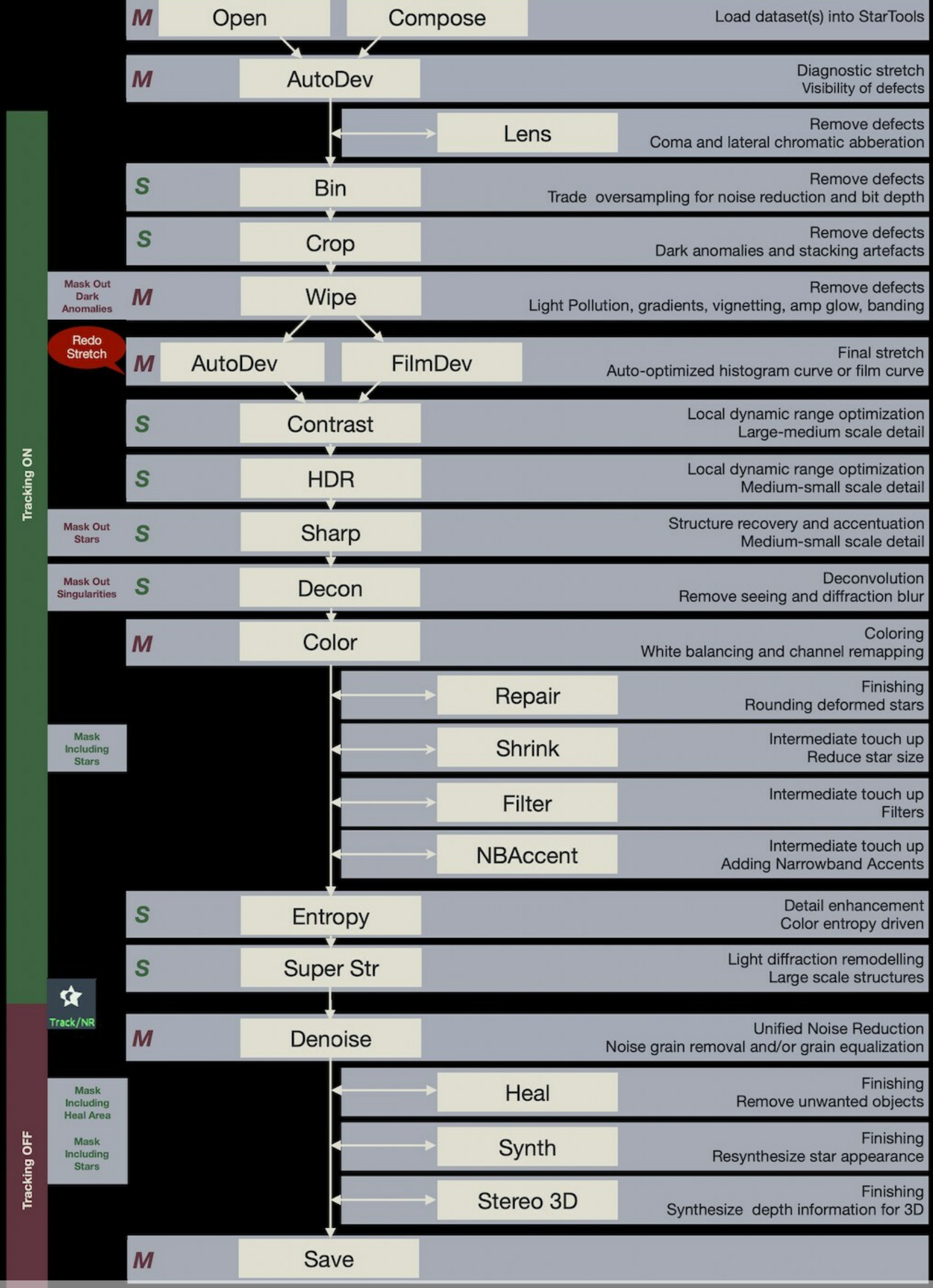
The icons in the top two panels roughly follow a recommended workflow when read left to right, top to bottom.

## Workflows

With a suitable dataset, workflows in StarTools are simple, replicable and short. Most modules are visited only once, with a clear purpose.

If you are familiar with other processing applications, you may be surprised with the seemingly erroneous mixing of modules that operate on linear vs non-linear data.

In StarTools, this important distinction is abstracted away, thanks to the signal evolution Tracking engine. In fact, it lets you do things, with ease, that are hard or impossible in other applications.




This excellent workflow chart by J. Scharmann shows a recommended core sequence of modules and actions some of which are (M)andatory, while others are merely (S)uggested. See links & tutorials section for more elaborate workflows.

## Step 1: Import, start Tracking

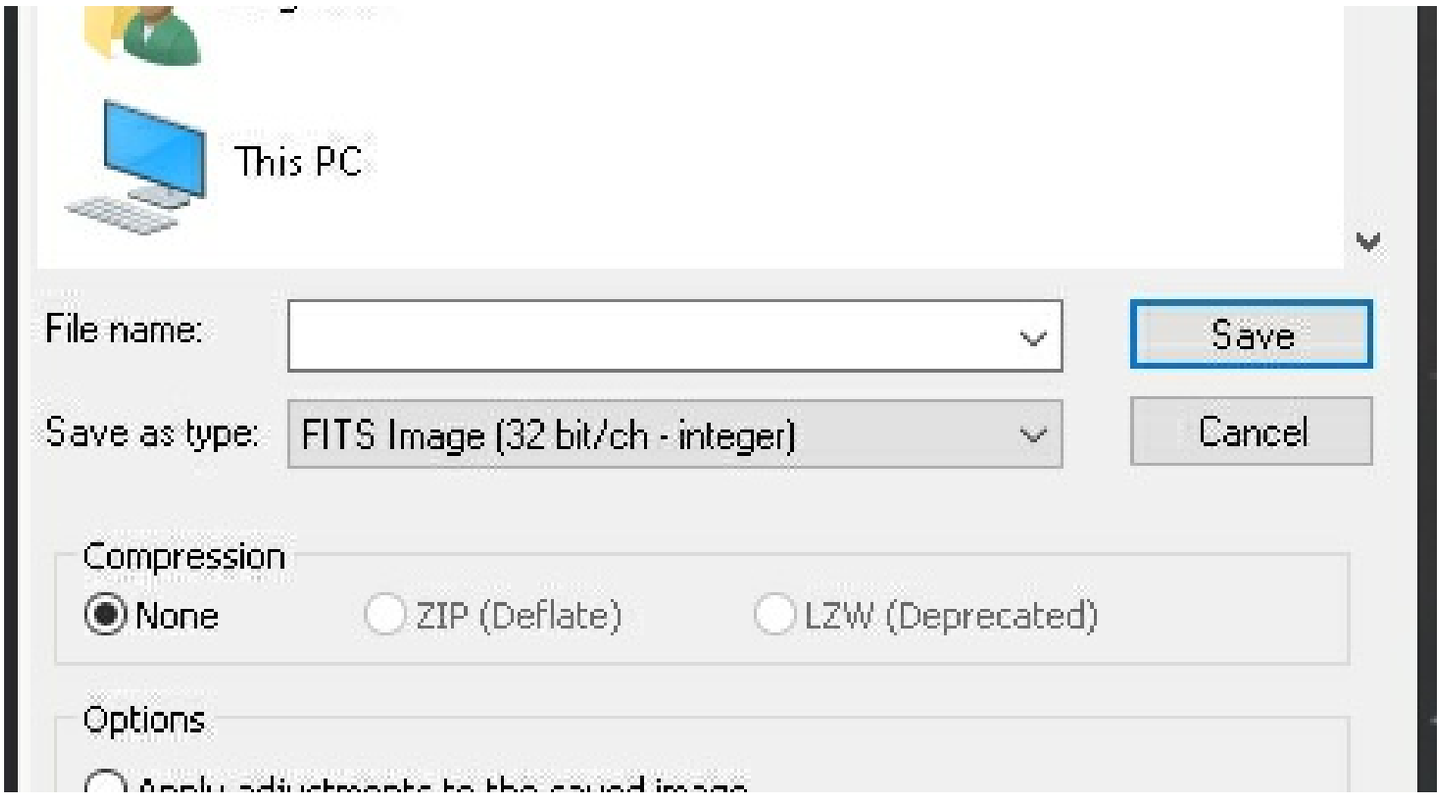
Open an image stack (''dataset''), fresh from a stacker. Make sure the dataset was stacked correctly, as StarTools, more than any other software, will not work (or work poorly) if the dataset is not stacked correctly or has been modified beforehand. Your dataset should be as ''virgin'' as possible, meaning unstretched, not colour balanced, not noise reduced and not deconvolved. Please consult the ''starting with a good dataset'' section in the ''links & tutorials'' section.

Upon opening an image, the Tracking dialog will open, asking you about the characteristics of the data. Choose the option that best matches the data being imported. If your dataset comes straight from a stacker, the first option is always safe. The second option may yield even better results if certain conditions are met. Depending on what you choose here, StarTools may work exclusively on the luminance (mono) part of your image, bringing in color later; StarTools is able to seamlessly process color and detail separately (yet simultaneously).

Tracking is now engaged (the Track button is lit up green). This means that StarTools is now monitoring how your signal (and its noise component) is transformed as you process it.

Once imported, counter-intuitively, a good stacker output will have a distinct, heavy color bias with little or no apparent detail. Worry not; subsequent processing in StarTools will remove the color bias, while restoring and bringing out detail. If, looking at the initial image, you are

wondering how on earth this will be turned into a nice picture, you are often on the right track.




Please make sure you calibrate, stack and save your images correctly. See the links & tutorials section for more information.

## Step 2: Inspect your dataset

Launch AutoDev to help inspect the data. Chances are that the image looks terrible, which is - believe it or not - the point. In the presence of problems, AutoDev will show them until they are dealt with. Because StarTools constantly tries to make sense of your data, StarTools is very sensitive to artefacts, meaning anything that is not real celestial detail (a single color bias, stacking artefacts, dust donuts, gradients, terrestrial scenery, etc.). Just 'Keep' the result. StarTools, thanks to Tracking, will allow us to redo the stretch later on.

At this point, things to look out for are;

- Stacking artefacts close to the borders of the image. These are dealt with in the Crop or Lens modules
- Bias or gradients (such as light pollution or skyglow). These are dealt with in the Wipe module.
- Oversampling (meaning the finest detail, such as small stars, being "smeared out" over multiple pixels). This is dealt with in the Bin module.
- Coma or elongated stars towards one or more corners of the image. These can be ameliorated using the Lens module.

Make mental notes of any issues you see.

Pre-colour balanced DSLR or OSC datasets like this modest one, will exhibit yellow, red or brown light pollution. However ideally you will want your stacker to not colour balance your dataset at all.

# Step 3

Step 3: Prep

Fix the issues that AutoDev has brought to your attention;

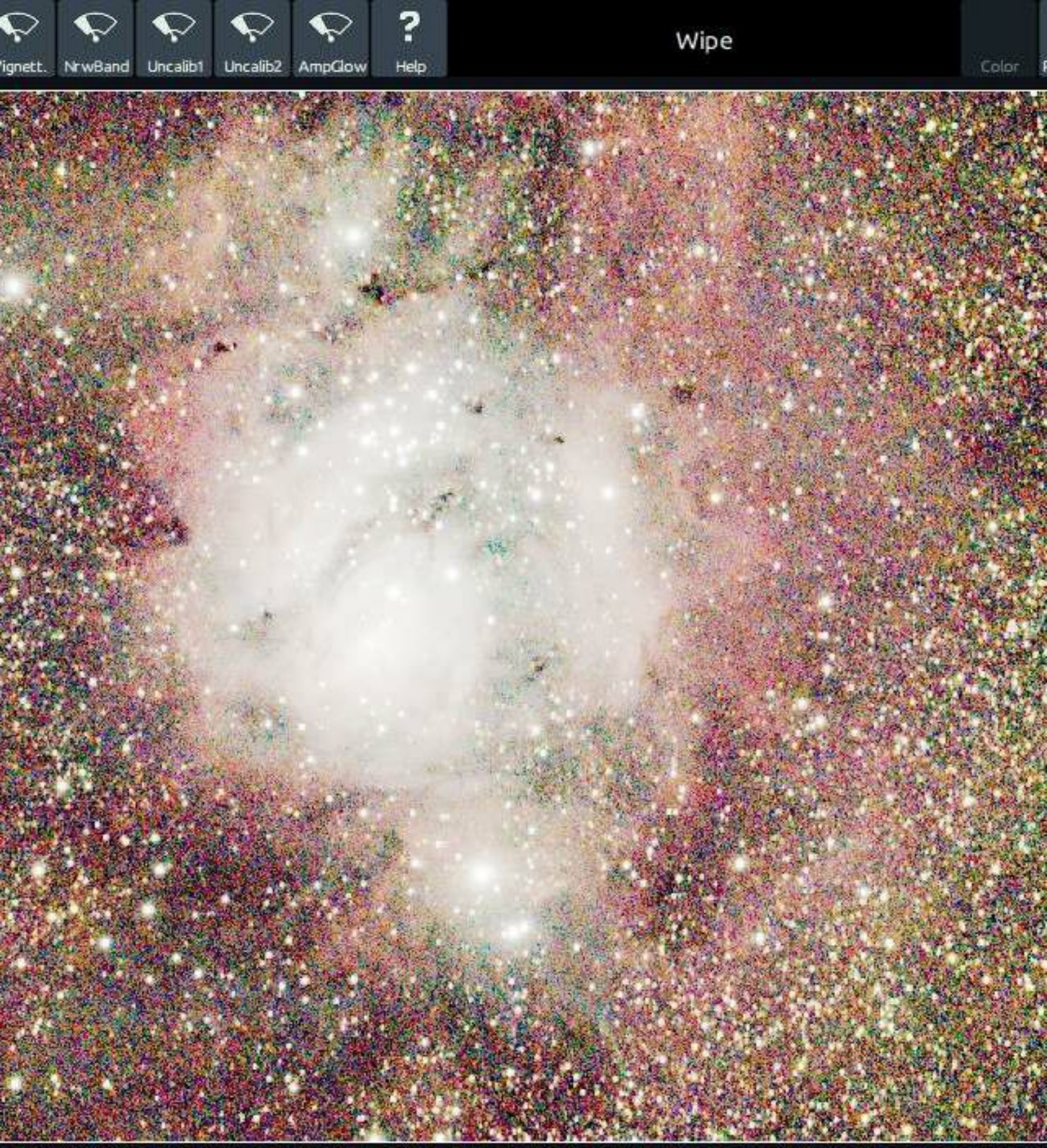
1 Ameliorate coma using the Lens module.

2 Crop any remaining stacking artefacts.

3 Bin the image up until each pixel describes one unit of real detail.

4 Wipe gradients and bias away. Be very mindful of any dark anomalies - bump up the Dark Anomaly filter if dealing with small ones (such as dark pixels) or mask big ones (such as large dust donuts) out using the Mask editor.

The importance of binning your dataset cannot be overstated. It will trade "useless" resolution for improved signal, making your dataset much quicker and easier to process, while allowing you to pull out more detail.




The Wipe module will keep showing you the warts in your data through a temporary, specialised 'diagnostics' stretch. The goal in Wipe is to clean up any gradients, vignetting and some other calibration defects.

## Step 4: Final global stretch

Once all issues are fixed, launch AutoDev again and tell it to 'redo' the stretch. If all is well, AutoDev will now create a histogram stretch that is optimised for the ''real'' object(s) in your cleaned-up dataset.

If your dataset is very noisy, it is possible AutoDev will optimise for the fine noise grain, mistaking it for real detail. In this case you can tell it to Ignore Fine detail.

If your object(s) reside on an otherwise uninteresting or ''empty'' background, you can tell AutoDev where the interesting bits of your image are by clicking & dragging a Region Of Interest (''RoI''). There is no shame in trying multiple RoIs. AutoDev will keep solving for a global strecth that best shows the detail in your RoI.

Understanding how AutoDev works is key to getting superior results with StarTools.

A second launch of AutoDev should show a much more reasonable image, free of gradients. Click & drag a region of interest ('RoI') to optimise the stretch for a specific area.

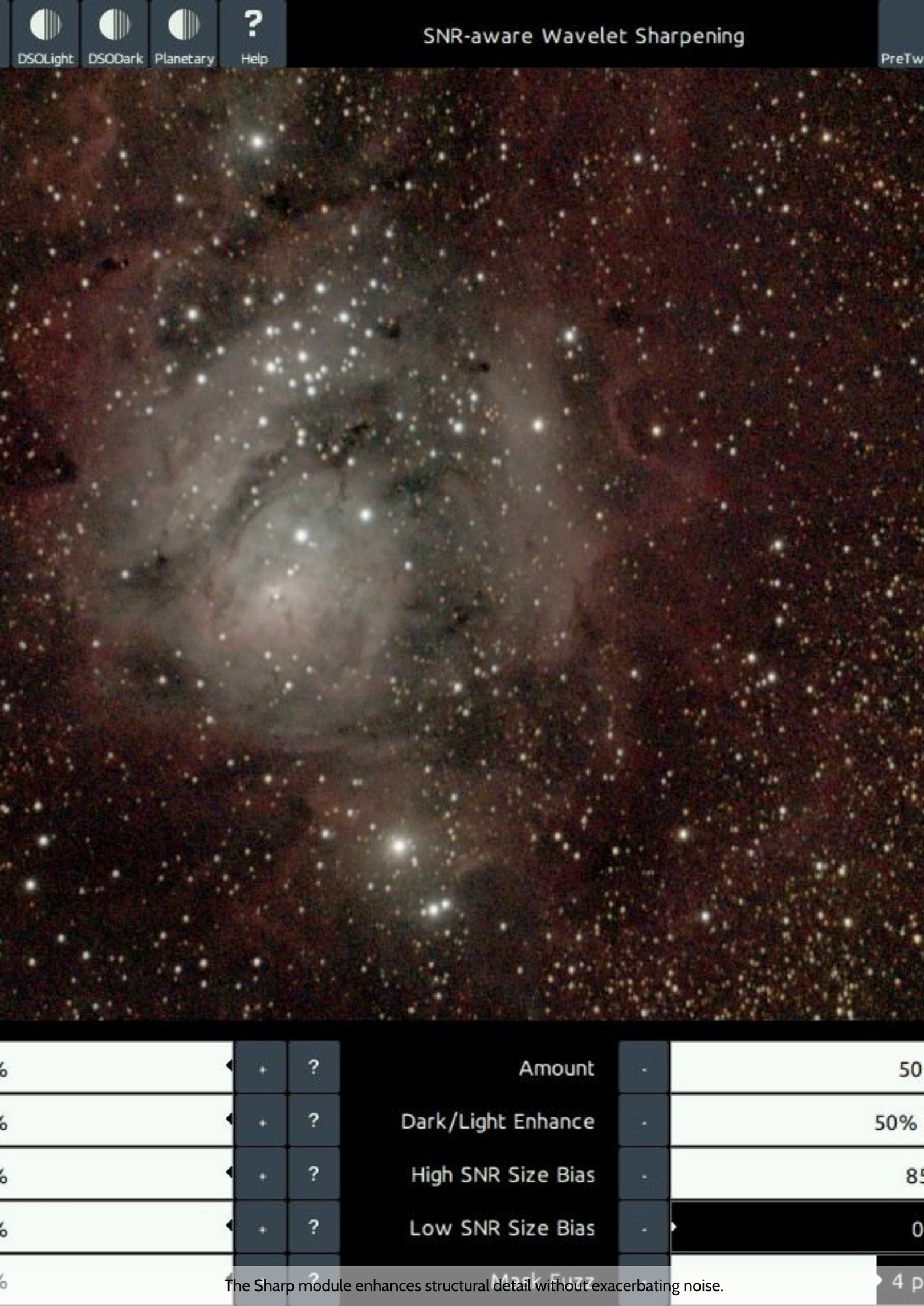
If even visible, don't worry about the colouring just yet - focus getting the detail out of your data first. If your image shows very bright highlights, know that you can "rescue" them later on using, for example, the HDR module.

## Step 5: Detail enhancement

Season your image to taste. Dig out detail with the Wavelet Sharpen ('Sharp') module, enhance Contrast with the Contrast module and fix any dynamic range issues with the HDR module.

Next, you can often restore blurred-out detail (for example due to an unstable atmosphere) using the easy-to-use Decon (deconvolution) module.

There are many ways to enhance detail to taste and much depends on what you feel is most important to bring out in your image. As opposed to other software, however, you don't need to be as concerned with noise grain propagation; StarTools will take care of noise grain when you finally switch Tracking off.




The Sharp module enhances structural detail without exacerbating noise.

## Step 6: Color calibration

Launch the Color module.

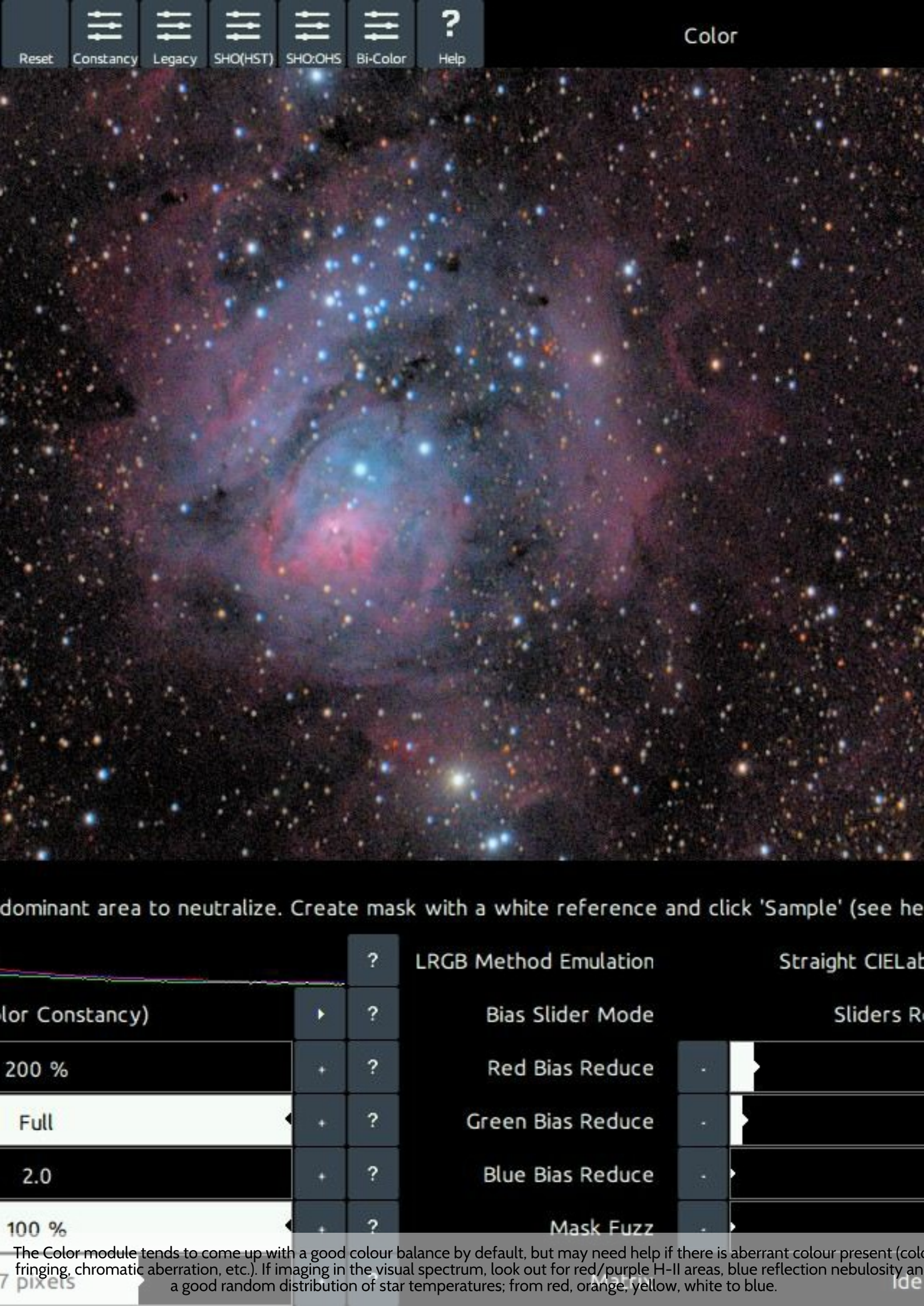
See if StarTools comes up with a good colour balance all by itself. A good colour balance shows a good range of all star temperatures, from red, orange and yellow through to white and blue. HII areas will tend to look purplish/pink, while galaxy cores tend to look yellow and their outer rims tend to look bluer.

Green is an uncommon colour in outer space (though there are notable exceptions, such as areas that are strong in OIII such as the core of M42). If you see green dominance, you may want to reduce the green bias. If you think you have a good colour balance, but still see some dominant green in your image, you can remove the last bit of green using the 'Cap Green' function.

StarTools is famous for its Color Constancy color rendering. This scientifically useful mode shows colours (for example nebula emissions) in the same color, regardless of brightness. However, if you prefer the more washed out and desaturated colour renderings of older software you can use the Legacy preset.

If your dataset has misaligned color channels or your optics suffer from chromatic aberration, the default colour balance may be off. Consult the Color module documentation for counter measures and getting a good colour balance.

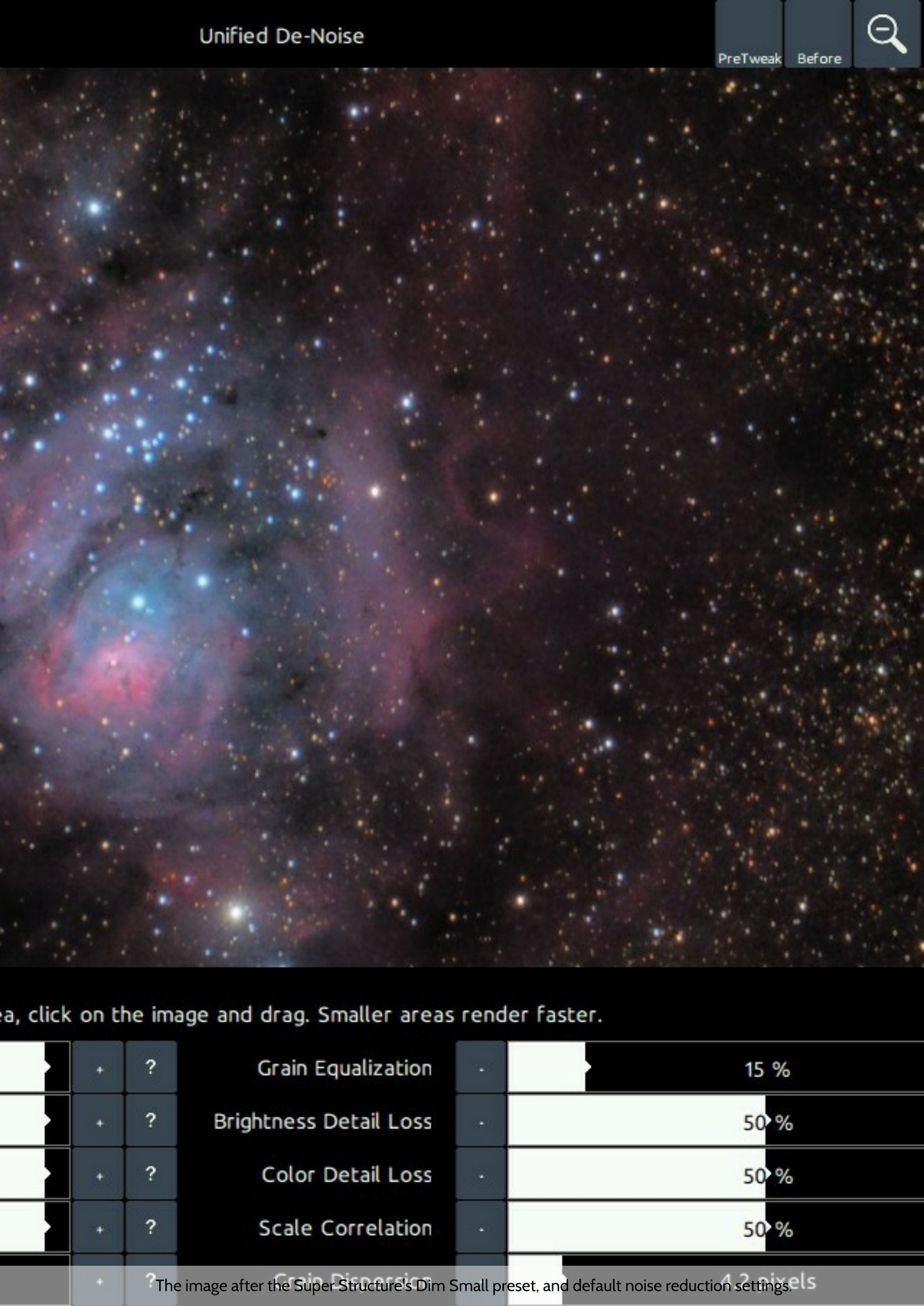
After colour calibration, you may wish to shrink stellar profiles, or use the Super Structure module ot manipulate the super structures relative to the rest of the image (for example to push back busy star fields).




The Color module tends to come up with a good colour balance by default, but may need help if there is aberrant colour present (colo
fringing, chromatic aberration, etc.). If imaging in the visual spectrum, look out for red/purple H-II areas, blue reflection nebulosity an
a good random distribution of star temperatures; from red, orange, yellow, white to blue.

## Step 7: Final noise reduction, switching Tracking off

Switch Tracking off and apply noise reduction. You will now see what all the "signal evolution Tracking" fuss is about, as StarTools seems to know exactly where the noise exists in your image, snuffing it out.




The image after the Super Structure's Dim Small preset, and default noise reduction settings.

## Step 8

Enjoy your final image!

If you find that, despite your best efforts, you cannot get a significantly better result in StarTools than in any (yes any!) other software, please contact us.

## Video

A video is also available that shows a simple, short processing workflow of a real-world, imperfect dataset.

Please refer to the video description below the video for the source data and other helpful links.

Watch video

https://www.youtube.com/watch?v=IDY3ov2dQe4

# 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.

In some modules a preview area can be specified in order to get a better idea of how settings would modify the image in a particular area, saving the user from waiting for the whole image to be re-calculated.

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, switching Tracking mode on/off, restoring the image to a particular state, 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 'Compose' module. Note also that some module may remain unavailable, depending on whether Tracking mode

is engaged.

Helpfully, the buttons are roughly arranged in a recommended workflow. Obviously not all modules need to be visited and workflow deviations may be needed, recommended or suit your personal taste better.

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.

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. A ''PreTweak/PostTweak'' button may also be available, which toggles between the current and previous result, allowing you to quickly spot the difference between two different settings.

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, situated to the right of the parameter control. These help buttons explain, again in brief, the nature of the parameter or setting.

Example of the main interface

The icons in the top two panels roughly follow a recommended workflow.

## Zooming, panning and scaling

Even the way StarTools displays and scales images, has been created specifically for astrophotography.

StarTools implements a custom scaling algorithm in its user interface, which makes sure that perceived noise levels stay constant, no matter the zoom level. This way, nasty noise surprises when viewing the image at 100% are avoided.

Even more clever, StarTools scaling algorithm can highlight latent and faint patterns (often indicating stacking problems or acquisition errors) by intentionally causing an aliasing pattern at different zoom levels in the presence of such patterns.

StarTools' astrophotography-optimised scaling algorithm can highlight latent pattern issues regardless of zoom level, as seen here. It also shows constant noise levels regardless of zoom level.

At 200% zoom level a barely distinguishable horizontal pattern can indeed be seen.

## Changing parameters in StarTools

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

1 A level setter, which allows the user to quickly set the value of a parameter within a certain range
2 An item selector, which allows the user to switch between different modes.

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.

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. Alternatively the user may click on the label area of the item selector to see the full range of items which may then be selected from a pop-over menu.

An example of a levelsetter control in StarTools

An example of a selector control in StarTools. Clicking on its center will reaveal all options as a pop-over menu.

## Presets

Most modules come with presets that quickly dial in useful parameter settings.

These presets give you good starting points for specific situations, and for basing your own tweaks on.

Preset buttons can be distinguished by their icons; they bear the icon of the module you launched. Most modules execute the first preset from the left by default upon opening.

Preset buttons can be distinguished by their icons; they bear the icon of the module you launched (in this case, the Wipe module)..

## Mouse controls

As of 1.7, enhanced mouse controls are implemented;

Zoom in

Scroll wheel down

Zoom out

Scroll wheel up

Pan

Middle button + drag

Blink before/after

Right click

## Hotkeys

As of version 1.5, StarTools implements some hotkeys for common functions;

Zoom out

- key

Zoom in

+ or = key

Zoom fit-to-screen

O key

Back

ESC key

Cancel

ESC key

Done

D or ENTER key

Keep

K key

OK

ESC key or ENTER key

Blink before / after

B key

Undo / redo

B key

Mask editor

### M key

Open

### O key

Save

### S key

Screenshot

### X key

## Touchscreen

StarTools can also be entirely operated by touchscreen with all controls appropriately sized for finger-touch operation.

# Tracking

Signal evolution Tracking data mining plays a very important role in StarTools and understanding it is key to achieving superior results with StarTools.

As soon as you load any data, StarTools will start Tracking the evolution of every pixel in your image, constantly keeping track of things like noise estimates, parameters you use and other statistics.

Tracking 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 the reason why you can do things in StarTools that would have 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!

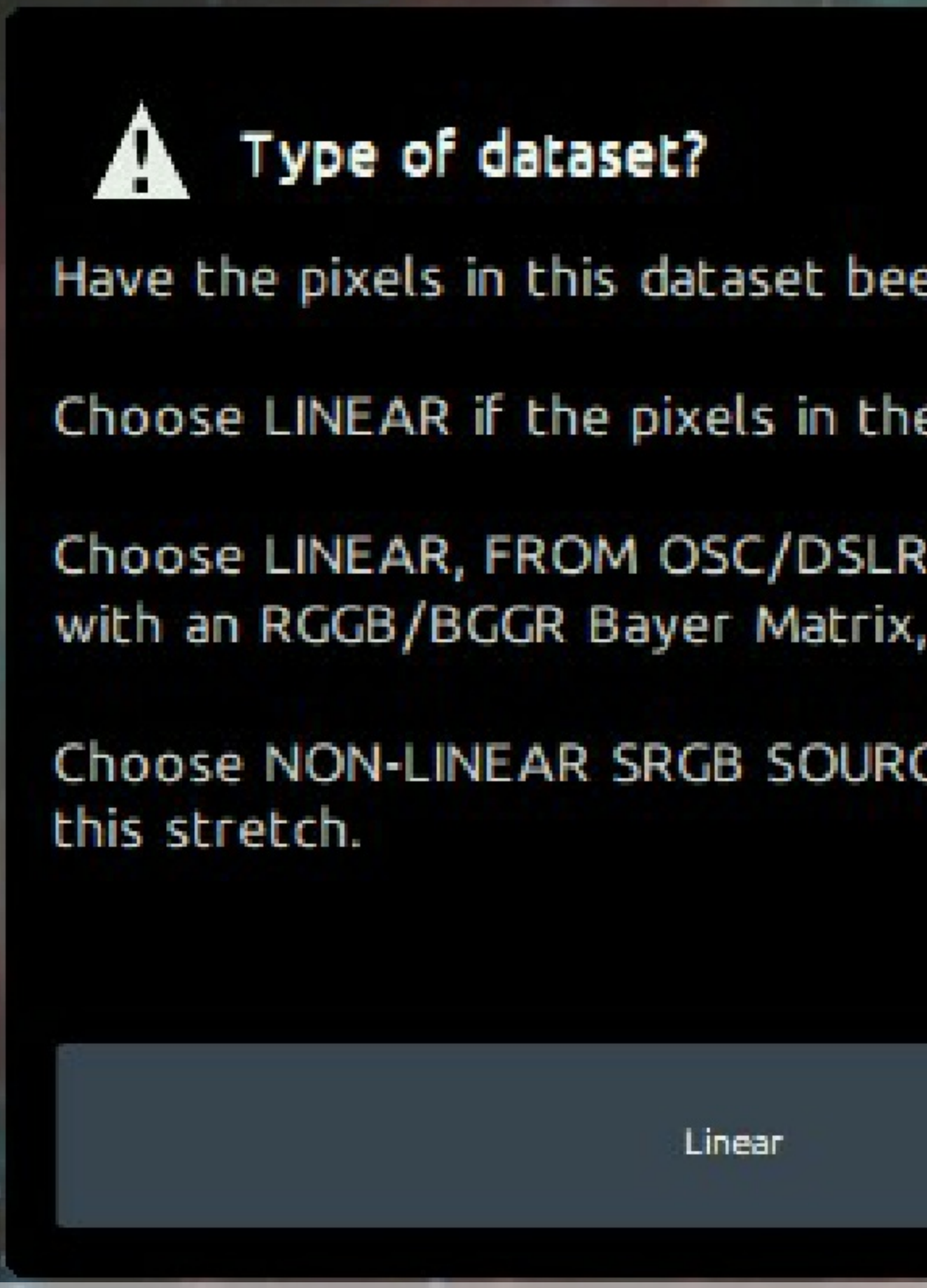
Tracking how you process your data also allows the noise reduction routines in StarTools to achieve superior results. By the time you get to your end result, the Tracking feature will have data-mined/pin-pointed exactly where (and how much) visible noise grain exists in your image. I therefore 'knows' exactly how much noise reduction to apply in each

area of your image.

Noise reduction is applied at the very end, as you switch Tracking off, because doing it at the very last possible moment will have given StarTools 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 colour calibration before doing any stretching, since it cannot deal with colour correction after the signal has been non-linearly transformed like StarTools can.

The knowledge that Tracking gathers is used in many other ways in StarTools, however, the nice thing about Tracking is that it is very unobtrusive. In fact, it actually helps get you get better results from your data in less time by homing in on parameters in the various modules that it thinks are good defaults, given what Tracking has learnt about your data.




Signal evolution Tracking starts as soon as you load your dataset.

# Log

StarTools keeps a detailed log of what modules and parameters you used. This log file is located in the same folder as the StarTools executable and is named **StarTools.log**.

As of the 1.4 beta versions, this log also includes the mask you used, encoded in base64 format. See the documentation on masks on how to easily decode the base64 if needed.

# GPU Acceleration

In all modules, suitable heavy arithmetic is offloaded to your Graphics Process Unit (GPU).

GPUs offer enormous advantages in compute power, under the right circumstances.

Depending on your hardware configuration and module, speed-ups versus the CPU-only version can range from 3x - 20x.

## Compatibility

StarTools supports virtually all modern GPUs and iGPUs on all modern Operating Systems.

StarTools is compatible with any GPU drivers that support OpenCL 1.1 or later. Almost all GPU released after ~2012 should have drivers available that expose this API.

StarTools GPU acceleration has been successfully tested on Windows, macOS and Linux with the following GPU and iGPU solutions;

- Nvidia GT/GTS/GTX 400, 500, 600, 700, 800M, 900, 1000 series
- Nvidia RTX series

- AMD HD 6700 series, HD 7800 series, HD 7900 series,R7 series, R9 series, RX series
- Intel HD 4000, HD 5000, UHD 620, UHD 630

Please note that if you card's chipset is not listed, StarTools may still work. If it does not (or does not do so reliably), please contact us.

## If you run into instabilities

Not all GPUs, operating systems and GPU drivers are created equal.

Some more consumer-oriented operating systems (e.g Windows, macOS), by default, assume the GPU is only used for graphics processing and not for compute tasks. If some compute tasks do not complete quickly enough, some drivers or operating systems may assume a GPU hang, and may reset the driver. This can particularly be an issue on systems with a relatively underpowered GPU (or iGPU) solution in combination with larger datasets. Please see the FAQ section on how to configure your operating system to mimimise this problem. Alternatively, you may consider using the CPU-only version.

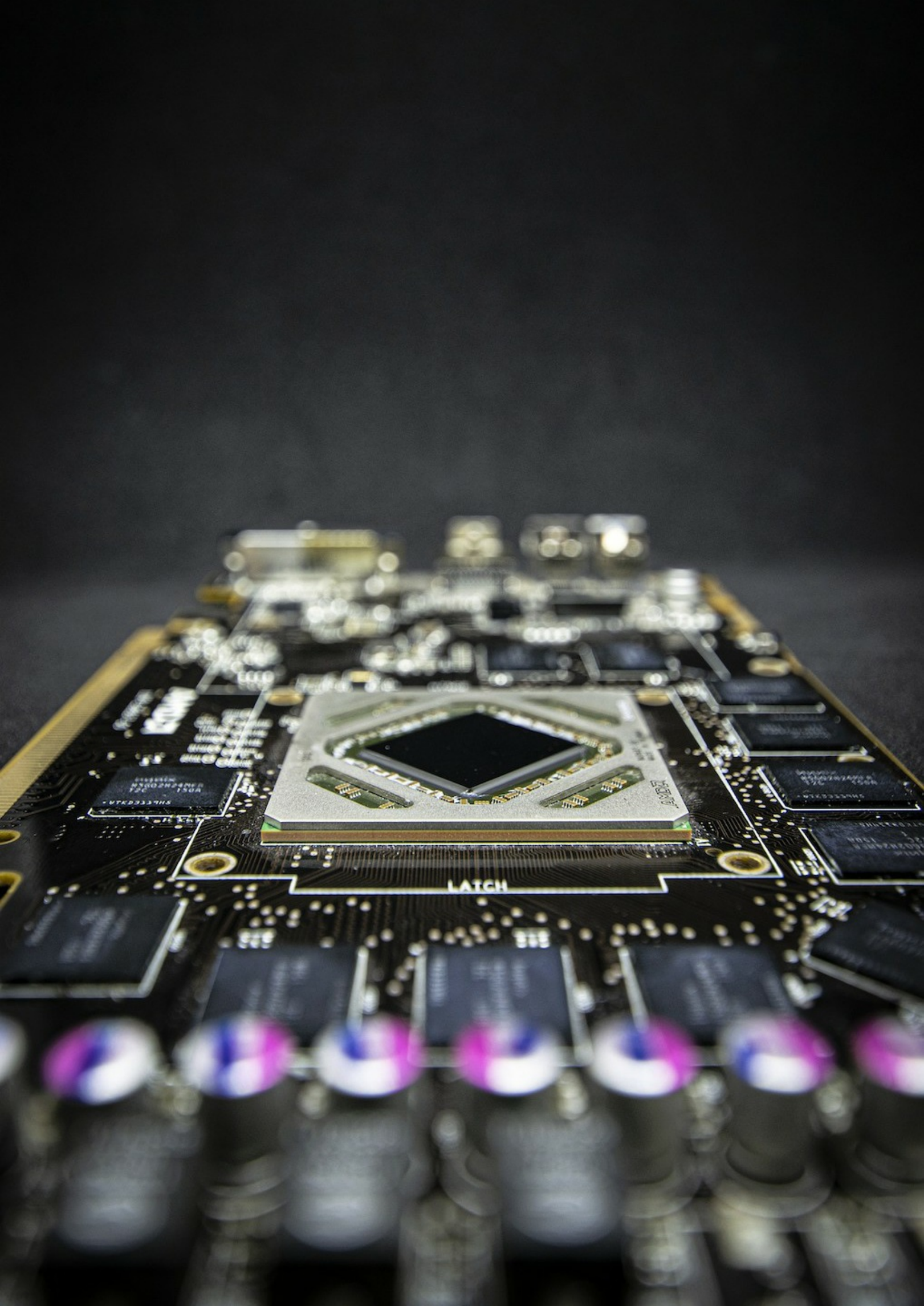
StarTools' algorithms push hardware to the limit and your GPU is no exception. If your GPU or power supply is ageing, StarTools will quickly lay bare weaknesses in thermal and power management. Similarly, laptops with iGPUs or discrete GPUs will have to work harder to rid themselves of waste heat.

## Burst loads versus sustained loads

Depending on your GPU monitoring application, it may appear your GPU is only used partially. This is not the case; your GPU solution is used and loaded up 100% where possible. However, as opposed to other tasks like video rendering or gaming, GPU usage in image processing tends to happens in *short*, but *very intense* bursts.

Depending on how your monitoring application measures GPU usage, these bursts may be too short to register. Spikes are averaged out over time by many monitoring applications. With the GPU loaded only for short times, but the load averaged out over longer periods, many monitoring applications make it appear only partial usage is happening.

If your monitoring application can show maximum values (on Windows you can try GPU-Z or Afterburner, on Linux the Psensor application), you should immediately see the GPU being maxed out. For examples of heavy sustained GPU activity, try the Deconvolution module with a high number of iterations or the Super Structure module.


LATCH

# Masks

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.

Importantly, 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.

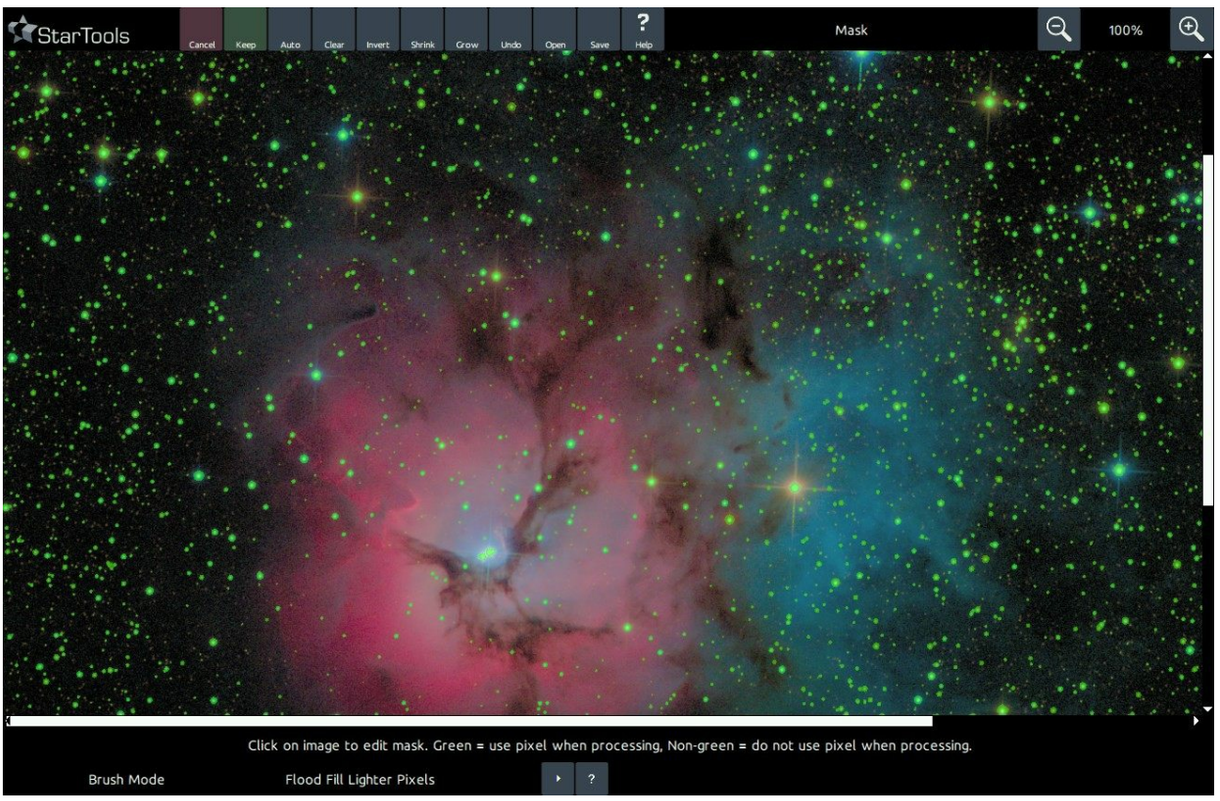
Green coloured pixels in the mask are considered 'on'. That is to say, they will be altered/used by whatever processing is carried out by the module you chose. 'Off' pixels (shown in their original colour) will not be altered or used by the active module. Again, please note that, by default all pixels 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 colour), then tell the module to 'keep the pixel as-is, hands off, do not touch or consider'.




Masking is an integral part of working with StarTools.

# Usage

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

Apart from different brush modes to set/unset pixels in the mask, 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 that shows 'on' pixels in pure white and 'off' pixels in pure black.

- 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. Loading an image that has values between black and white will designate any shades of gray closest to white as 'on', and any shades of gray closest to black as 'off'.
- The 'Auto' button is a very powerful feature that allows you to automatically isolate features.
- 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.
- 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.

**NOTE: To quickly turn on all pixels, click the 'clear' button, then the 'invert' button.**

## Brush modes

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. Click anywhere on a clump of 'on' (green) pixels, to toggle the whole clump off again.

The mask editor has 10 'Brush modes';

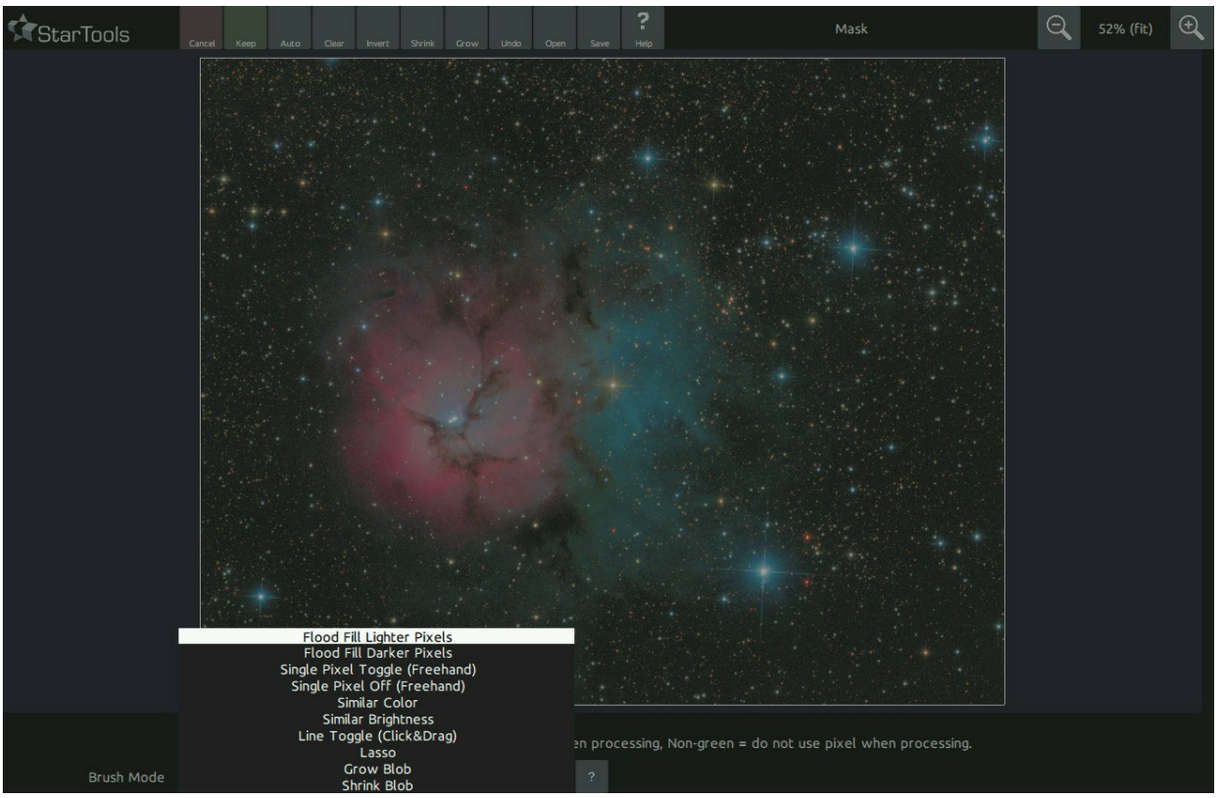
- **Flood fill lighter pixels**; use it to quickly select an adjacent area that is lighter than the clicked pixel (for example a star or a galaxy). Specifically, Clicking a non-green pixel will, starting from the clicked pixel, recursively fill the image with green pixels until it finds that; either all neighbouring pixels of a particular pixel are already filled (on/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 fill darker pixels**; use it to quickly select an adjacent area that is darker than the clicked pixel (for example a dust lane). Specifically, clicking a non-green pixel will, starting from the clicked pixel, recursively fill the image with green pixels until it finds that; either all neighbouring pixels of a particular pixel are already filled (on/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 on/green neighbouring pixels.

- **Single pixel toggle**; clicking a non-green pixel will make a non-green pixel turn green. Clicking a green pixel will make green pixel turn non-green. It is a simple toggle operation for single pixels.
- **Single pixel off (freehand)**; clicking or dragging while holding the mouse button down will turn off pixels. This mode acts like a single pixel "eraser".
- **Similar color**; use it to quickly select an adjacent area that is similar in color.
- **Similar brightness**; use it to quickly select an adjacent area that is similar in brightness.
- **Line toggle (click & drag)**; use it to draw a line from the start point (when the mouse button was first pressed) to the end point (when the mouse button was released). This mode is particularly useful to trace and select satellite trails, for example for healing out using the Heal module.
- **Lasso**; toggles all the pixels confined by a convex shape that you can draw in this mode (click and drag). Use it to quickly select or deselect circular areas by drawing their outline.
- **Grow blob**; grows any contiguous area of adjacent pixels by

expanding their borders into the nearest neighbouring pixel. Use it to quickly grow an area (for example a star core) without disturbing the rest of the mask.

- **Shrink blob**; shrinks any contiguous area of adjacent pixels by withdrawing their borders into the nearest neighbouring pixel that is not part of a border. Use it to quickly shrink an area without disturbing the rest of the mask.




10 different mask brush modes are at your disposal, to help you quickly create or touch up the mask you need

## The Auto Feature

The powerful 'Auto' function quickly and autonomously isolates features of interest such as stars, noise, hot or dead pixels, etc.

For example, isolating *just* the stars in an image is a necessity for obtaining any useful results from the 'Decon' and 'Magic' module.

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

- **Light features + highlight > threshold**; 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') and depending on a certain sensitivity (see 'Filter sensitivity').
- **Highlight > threshold**; selects any pixel whose brightness is brighter than a certain percentage of the maximum (e.g. pure white) value. . 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 color/mono < 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). Two versions are available, one for color images, the other for mono images.
- **Hot pixels color/mono > 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. Two versions are

available, one for color images, the other for mono images.

- **Noise**; 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.
- **Dust & scratches**; selects small specks of dusts and scratches as found on old photographs. Only the 'Threshold' parameter is used, and a very low value for the 'Threshold' parameter is needed.
- **Edges > Threshold**; 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.
- **Horizontal artifacts**; selects horizontal anomalies in the image. Use the 'Max feature size' and 'Filter sensitivity' to throttle the aggressiveness with which the detector detects the anomalies.
- **Vertical artifacts**; selects vertical anomalies in the image. Use the 'Max feature size' and 'Filter sensitivity' to throttle the aggressiveness with which the detector detects the anomalies.
- **Radius**; selects a circle, starting from the centre of the image going outwards. The 'Threshold' parameter defines the radius of the circle, where 100.00 covers the whole image.

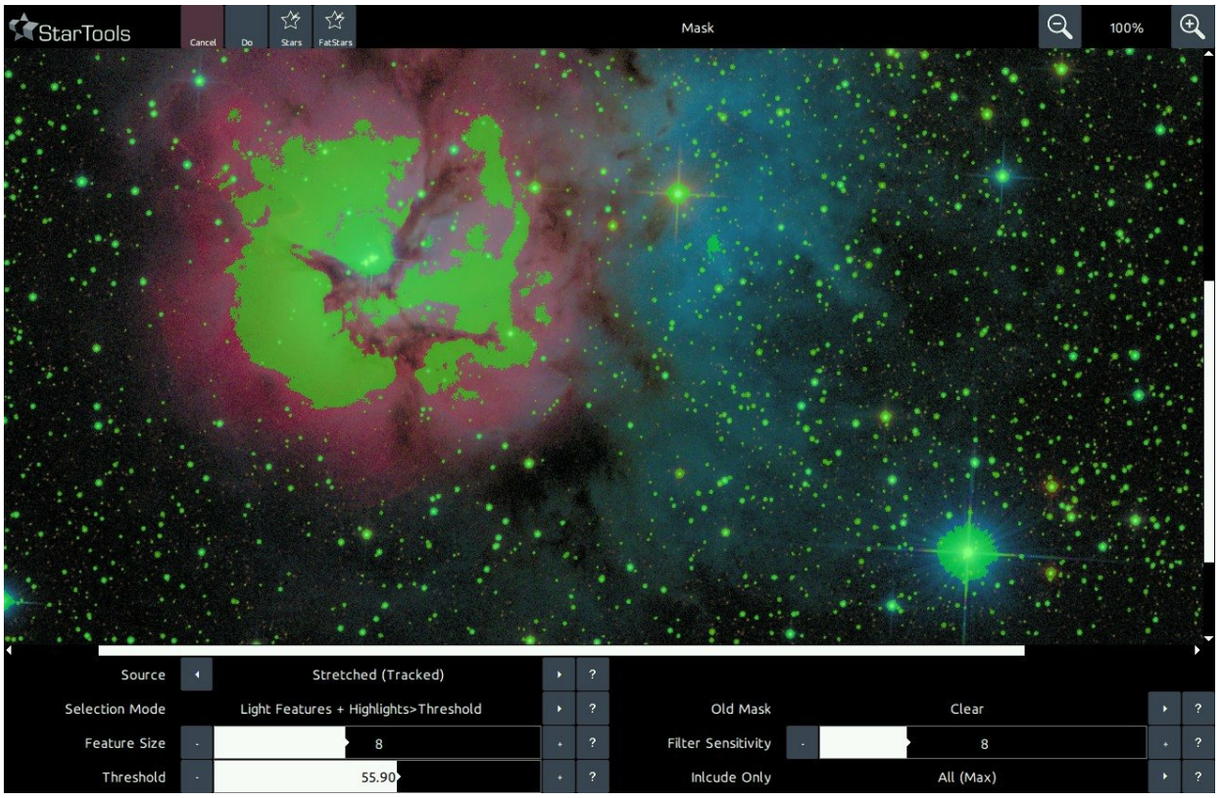
Some of the selection algorithms are controlled by additional

parameters;

- **Include only**; tells the selection algorithms evaluate specific colour channels only 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 setting 'Include only' to 'Green', you are able to tell the selection algorithms to leave red and blue 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.

Finally, the 'Source' parameter selects the source data the Auto mask generator should use. Thanks to StarTools' Tracking functionality which

gives every module the capability to go ''back in time'', the Auto mask generator can use either the original 'Linear' data (perfect for getting at the brightest star cores), the data as you see it right now ('Stretched'), or the data as you see now but taking into account noise propagation ('Stretched (Tracked)'). The latter greatly helps reduce false positives caused by noise.




The Auto Mask Generator is indispensible when, for example, dealing with star mask, as required for many of the modules in StarTools.

# Using masks from startools.log

StarTools stores the masks you used in your workflow in the StarTools.log file itself. This StarTools.log file is located in the same folder as the executables. The masks are encoded as BASE64 PNG images. To convert the BASE64 text into loadable PNG images, you can use any online (or offline) BASE64 converter tool.

The part to copy and paste, typically starts with;

```
iVBOR.....
```

## Online BASE64 converter by Motobit

One online tool for BASE64 is Motobit Software's BASE64 encoder/decoder.

To use it to convert StarTools masks back into importable PNG files;

- Paste the BASE64 code into the text box
- Select the **'decode the data from a Base64 string (base64 decoding)'** radio button
- Select the '**export to a binary file, filename:**' radio button.
- Name the file for example "mask.png"
- Click the **convert the source data** button.

This should result in a download of the mask as a PNG file which can be imported into the StarTools mask editor, as welll as other applications.

Online BASE64 converter by Motobit Software

https://www.motobit.com/util/base64-decoder-encoder.asp

## Advanced techniques

The mask editor and its auto-mask generator are very flexible tools. These more advanced techniques will allow you to create specialised masks for specific situations and purposes.

### Object protection

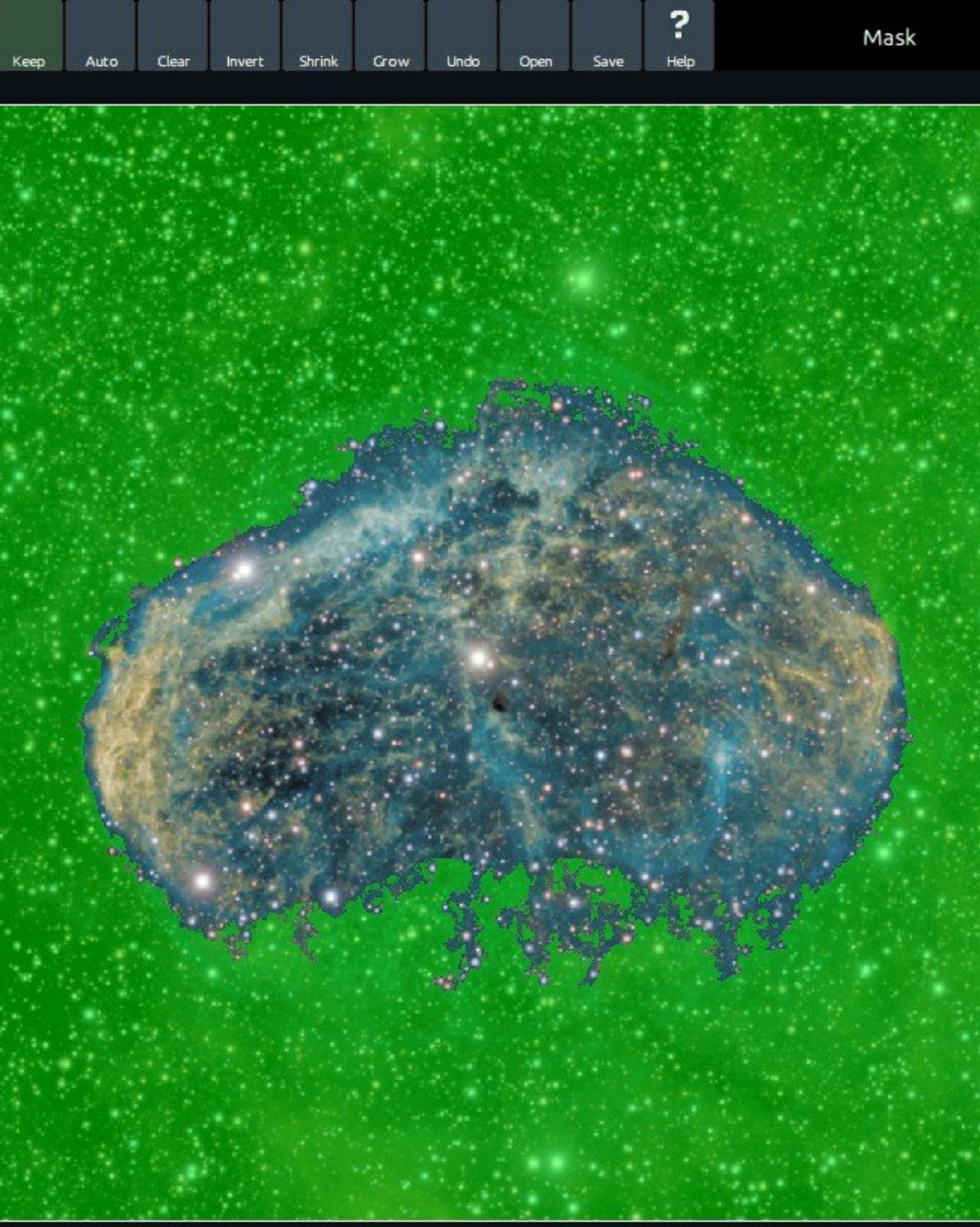
Sometimes, it is desirable to keep an object or area from being included in an auto-generated mask. It is possible to have the auto-mask generator operate only on designated areas;

1 Clear the mask, and select the part of the image

you wish to protect with the Flood Fill Lighter or Lasso tool, then click Invert.

2 In the Auto mask generator, set the parameters you need to generate your mask. Be sure to set 'Old Mask' to 'Add New Where Old Is Set'.

3 After clicking 'Do'. The auto-generator will generate the desired mask, however excluding the area we specified earlier.




Select the part of the image you wish to protect with the Flood Fill Lighter or Lasso tool, then click Invert..

In the Auto mask generator, set the parameters you need to generate your mask. Be sure to set 'Old Mask' to 'Add New Where Old Is Set'.

After clicking 'Do'. The auto-generator will generate the desired mask, however excluding the area we specified earlier.

## The ethics of using masks and selective processing

Where documentary photography is concerned, selective manipulation *by hand* is typically frowned upon, unless the practice of it is clearly stated when the final result is presented.

However, in cases where a mask is algorithmically derived, purely from the dataset itself, without adding any outside extra information, masking is common practice even in the realm of documentary photography. Examples of such use cases are range masks (for example, selecting highlights only based on brightness), star mask (selecting stars only based on stellar profile), colour masks (selecting features based on colour), etc.

In some modules in StarTools specifically, masks are used for the purpose of selective sampling to create internal parameters for an operation that is applied *globally* to all pixels. This too is common practice in the realm of documentary photography. Examples of such use cases are gradient modelling (selecting samples to model a global gradient on) and color balancing (selecting samples to base an global white balance on).

Finally, it is also generally permitted to mask out singularities (pixels with a value that is unknown) by hand, in order to exclude this from some operations that may otherwise generate artefacts in response to encountering these. Examples may be over-exposing star cores, dead or hot pixels, stacking artefacts, or other data defects.

As a courtesy, when in doubt, it is always good to let you viewers know how you processed an image, in order to avoid confusion.

# AutoDev

AutoDev is an advanced image stretching solution that relies on detail analysis, rather than on the simple non-linear transformation functions from yesteryear.

To be exact, in StarTools, Histogram Transformation Curves (DDP, Levels and Curves, ArcSinH stretch, MaskedStretch etc.) are considered obsolete an non-optimal; AutoDev uses robust, controllable image analysis to achieve better, more objective results in a more intuitive way.

When data is acquired, it is recorded in a linear form, corresponding to raw photon counts. To make this data suitable for human consumption, stretching it non-linearly is required. Historically, simple algorithms were used to emulate the non-linear response of photographic paper by modelling its non-linear transformation curve. Later, in the 1990s because dynamic range in outer space varies greatly, "levels and curves" tools allowed imagers to create custom histogram transformation curves that better matched the object imaged so that the most amount of detail became visible in the stretched image.

Creating these custom curves was a highly laborious and subjective process. And, unfortunately, in many software packages this is still the situation today. The result is almost always sub-optimal dynamic range allocation, leading to detail loss in the shadows (leaving recoverable

detail unstretched), shrouding interesting detail in the midtones (by not allocating it enough dynamic range) or blowing out stars (by failing to leave enough dynamic range for the stellar profiles). Working on badly calibrated screens, can exacerbate the problem of subjectively allocating dynamic range with more primitive tools.

StarTools' AutoDev module uses image analysis to find the optimum custom curve for the characteristics of the data. By actively looking for detail in the image, AutoDev autonomously creates a custom histogram curve that best allocates the available dynamic range to the scene, taking into account all aspects and detail. As a consequence, the need for local HDR manipulation is minimised.

AutoDev is in fact so good at its job, that it is also one of the most important tools in StarTools for initial data inspection. Using AutoDev as one of the first modules on your data will see it bring out problems in the data, such as stacking artifacts, gradients, bias, dust donuts, and more. Precisely per its design goal, its objective dynamic range allocation will bring out such defects so these may be corrected, or at the very least taken into account by you during processing.

Upon removal and/or mitigation of these problems, AutoDev may then be used to stretch the cleaned up data, bringing out detail across the entire dynamic range equally.

Top: traditional Digital Development curve (via FilmDev module), Bottom: AutoDev. Notice the vastly better dynamic range allocation, with more detail visible in the shadows and highlights, while not compromising on detail in midtones or blowing out stars. The AutoDev image is the perfect starting point for enhancing local detail.

Not a bug, but a feature! Don't let a first result like this scare you. AutoDev is doing you a favor by showing you exactly what is wrong with your data. In this we can see heavy light pollution, gradients and stacking artifacts that need taking care of before we can go any further.

# Usage

AutoDev is used for two distinct purposes;

1 To visualise artifacts and problems in your dataset.

2 To stretch the real celestial signal in your dataset

Using AutoDev is typically one of the first things a StarTools user does. This is because AutoDev, in the presence of any issues, brings out those issues, just like it would with real detail. Any such issues, for example stacking artifacts, gradients, dust donuts, noise levels, oversampling, etc., can then first be addressed by the relevant modules.

Once the issues have been dealt with to the best of your ability, AutoDev can be used again to stretch your final image to visualise the detail (rather than any artifacts). Do not attempt to use AutoDev for the purpose of bringing out detail if you have not taken care of forementioned artifacts and issues.

## Improvements over basic histogram stretching

To be able to detect detail, AutoDev has a lot of smarts behind it. Its main detail detection algorithm analyses a Region of Interest ("RoI") - by default the whole image - so that it can find the optimum histogram transformation curve based on what it "sees".

Understanding AutoDev on a basic level is pretty simple really; its goal is to look at what's in your image and to make sure as much as possible is visible, just as a human would (try to) look at what is in the image and approximate the optimal histogram transformation curves using traditional tools.

The problem with a histogram transformation curve (aka 'global stretch') is that it affects all pixels in the image. So, what works in one area (bringing out detail in the background), may not necessarily work in another (for example, it may make a medium-brightness DSO core harder to see). Therefore it is important to understand that - fundamentally - globally stretching the image is *always* a compromise. AutoDev's job then, is to find the best-compromise global curve, given what detail is visible in your image and your preferences. Of course, fortunately we have other tools like the Contrast, Sharp and HDR modules to 'rescue' *all* detail by optimising for local dynamic range on top of global dynamic range.

Being able to show all things in your image equally well, is a really useful feature, as it is *also* very adept at finding artefacts or stuff in your image that is **not** real celestial detail but requires attention. That is why AutoDev is also extremely useful to launch as the first thing after loading an image to see what - if any - issues need addressing before proceeding. If there are any, AutoDev is virtually guaranteed to show them to you. After fixing such issues (for example using Crop, Wipe or

other modules), we can go on to use AutoDev's skills for showing the remaining (this time *real celestial*) detail in the image.

If most of the image consists of a background and just a small object of interest, by default AutoDev will weigh the importance of the background higher (since it covers a much larger part of the image vs the object). This is understandable and neatly demonstrates its behavior. It will always look for the best compromise stretch to show the entire **Region of Interest** (''RoI'' - by default the entire image). This also means that if the background is noisy, it will start digging out the noise, taking it as ''fine detail'' that needs to be ''brought out''. If this behaviour is undesirable, there are a couple of things you can do in AutoDev.

1 Change the **'Ignore Fine Detail <'** parameter, so that AutoDev will no longer detect fine detail (such as noise grain).

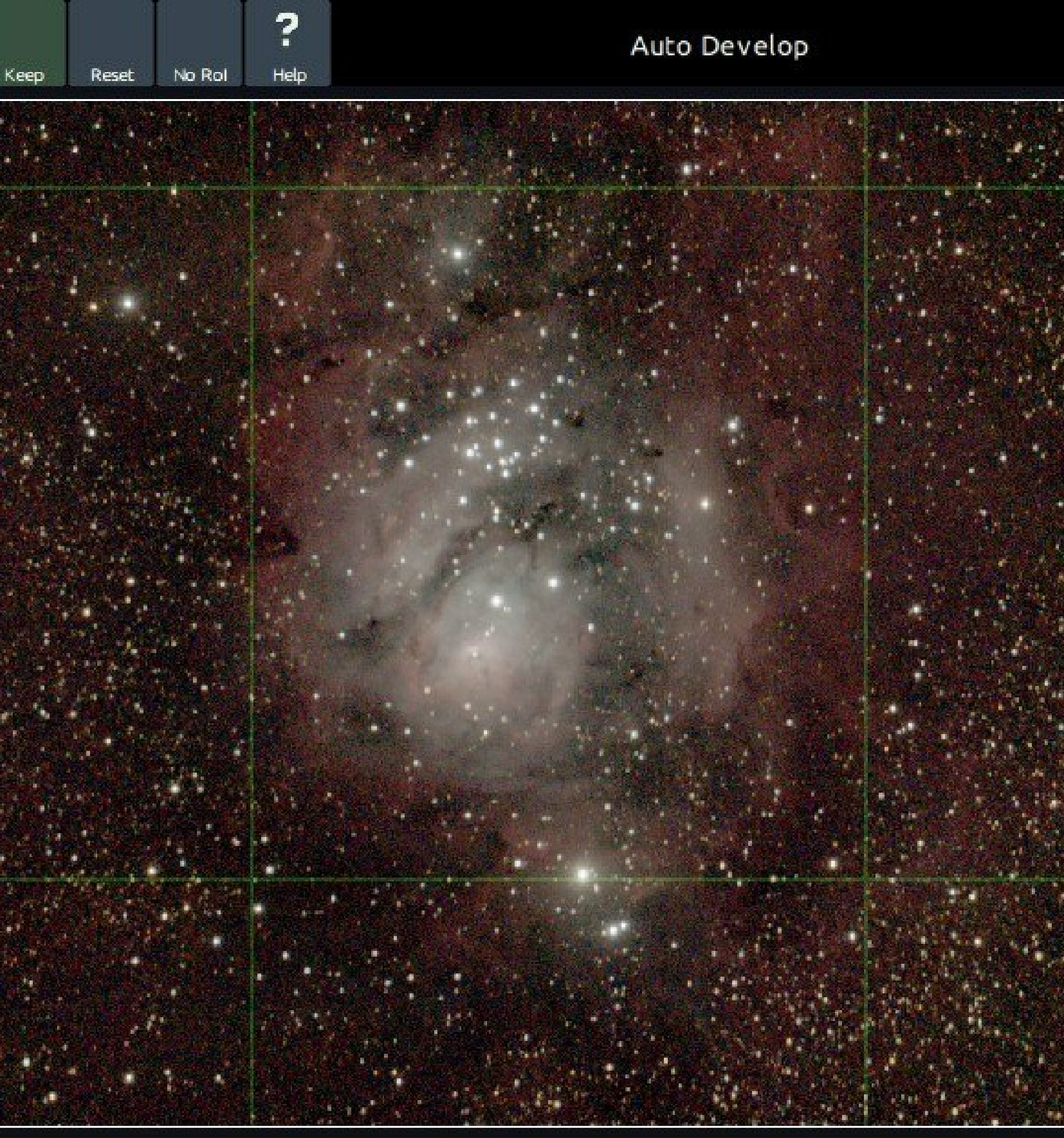
2 Simply tell it what it should focus on instead by specifying an ROI and not regard the area outside the ROI just a little bit ('**Outside ROI influence**').

You will find that, as you include more background around the object, AutoDev, as expected, starts to optimise more and more for the background and less for the object. To use the RoI effectively, give it a ''sample'' of the important bit of the image. This can be a whole object, or it can be just a slice of the object that is a good representation of what's going on *in* the object in terms of detail. You can, for example, use

a slice of a galaxy from the core, through the dust lanes, to the faint outer arms. There is no shame in trying a few different ROIs in order to find one you're happy with. What ever the case, the result will be more optimal and objective than pulling at histogram curves.

There are two ways of further influencing the way the detail detector "sees" your image;

- The '**Detector Gamma**' parameter applies - for values other than 1.0 - a non-linear stretch to the image prior to passing it to the detector. E.g. the detector will "see" a darker or brighter image and create a curve that suits this image, rather than the real image. This makes the detector proportionally more (< 1.0) or less (> 1.0) sensitive to detail in the highlights. Conversely it makes the detector less (<1.0) or more (> 1.0) sensitive to detail in the shadows. The effect can be though of as a "smart" gamma correction. Note that tweaking this parameter will, by virtue of its skewing effect, cause the resulting stretch to no longer be optimal.
- The '**Shadow Linearity**' parameter specifies the amount of linearity that is applied in the shadows, before non-linear stretching takes over. Higher amounts have the effect of allocating more dynamic range to the shadows and background.




Great allocation of dynamic range by AutoDev after taking care of the stacking artifacts, gradients and light pollution using the Wipe module.

## Understanding AutoDev's behavior

In AutoDev, you are controlling an impartial and objective detail detector, rather than a subjective and hard to control (especially in the highlights) bezier/spline curve.

Having something impartial and objective taking care of your initial stretch is very valuable, as it allows you to much better set up a "neutral" image that you can build on with the other *local* detail-enhancing tools in your arsenal (e.g. Sharp, HDR, Contrast, Decon, etc.). For example, when using Autodev, it will quickly become clear that point lights and over-exposed highlights, such as the cores of bright stars, remain much more defined. The dreaded "star bloat" effect is much less pronounced or even entirely absent, depending on the dataset.

However, knowing how to effectively use Region of Interests ("RoI") is crucial to making the most of AutoDev. Particularly if the object of interest is not image-filling, a Region of Interest will often be necessary. Fortunately the fundamental workings of the RoI are easy to understand.

Detail inside the RoI

Let's say our image is of galaxy, neatly situated in the center. Then confining the RoI progressively to the core of the galaxy, the stretch becomes more and more optimised for the core and less and less for the outer rim. Conversely, if we want to show more of the outer regions as well, we would include those regions in the RoI.

Detail outside the RoI

Shrinking or enlarging the RoI, you will notice how the stretch is optimised specifically to show as much as possible of the image *inside of* the RoI. That is not to say any detail *outisde* the RoI shall be invisible. It just means that any detail there will not (or much less) have a say in how the stretch is made. For example, if we would have an image of a galaxy, cloned it, put the two image side by side to create a new image, and then specified the RoI perfectly over just one of the cloned galaxies, the other one, outside the RoI would be stretched precisely the same way (as it happens to have exactly the same detail). Whatever detail lies outside the RoI, is simply forced to conform to the stretch that was designed for the RoI.

It is important to note that AutoDev will never clip your blackpoints outside the RoI, unless the '**Outside RoI Influence**' parameter is explicitly set to 0% (though it is still not guaranteed to clip even at that setting). Detail outside the RoI may appear very dark (and approach 0/black), but will never be clipped.

Bringing up the '**Outside RoI Influence**' parameter will let AutoDev allocate the specified amount of dynamic range to the area outside the RoI as well, at the expens of some dynamic range *inside* the RoI. If '**Outside RoI Influence**' set 100%, then precisely 50% of the dynamic range will be used to show detail inside the RoI and 50% of the dynamic range will be used to show detail outside the RoI. Note that, visually, this

behavior is area-size dependent; if the RoI is only a tiny area, the area outside the RoI will have to make do with just 50% of the dynamic range to describe detail for a much larger area (e.g. it has to divide the dynamic range over many more pixels), while the smaller RoI area has much fewer pixels and can therefore allocate each pixel more dynamic range if needed, in turn showing much more detail.

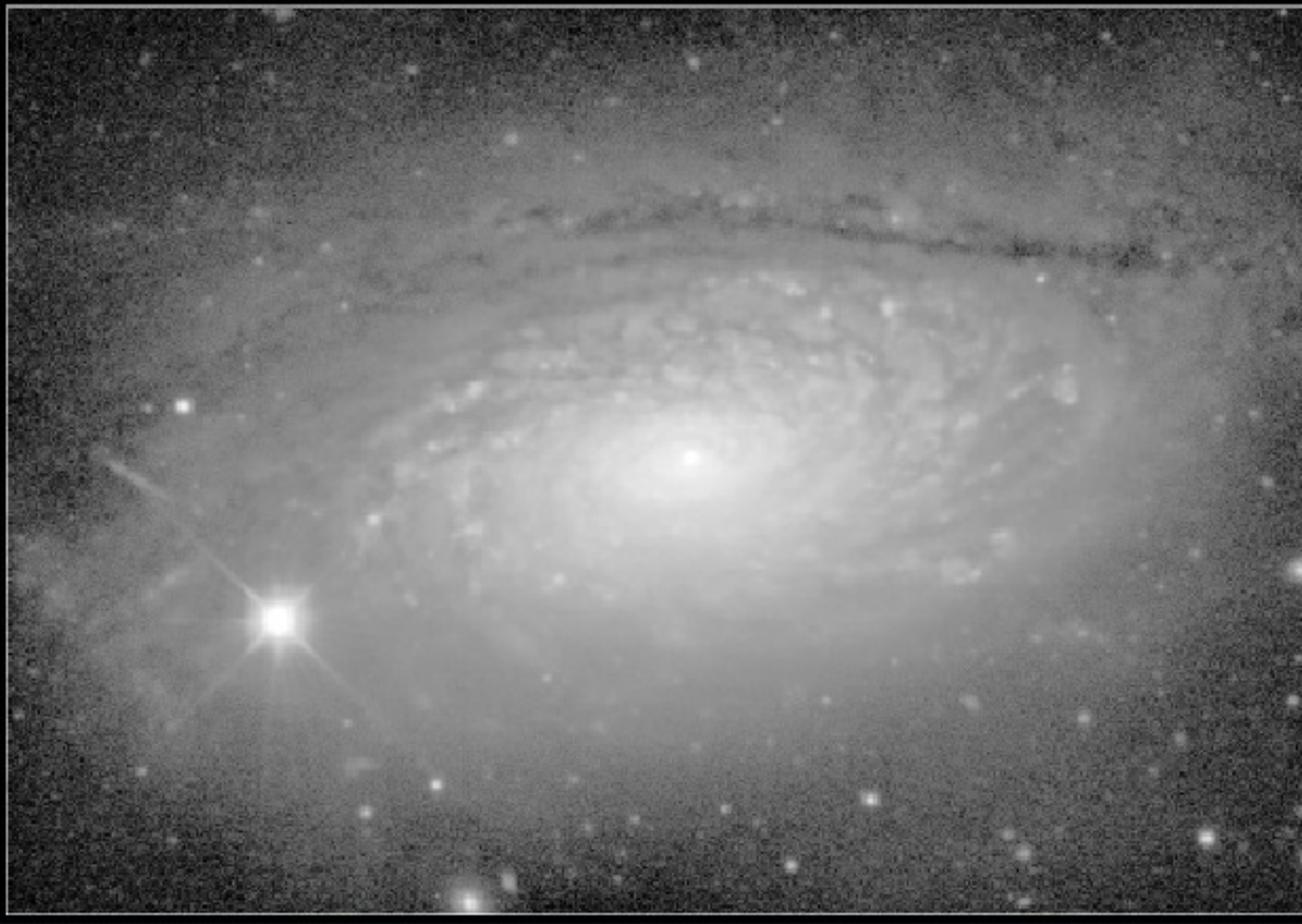
Choosing the RoI in case of multiple objects of interest

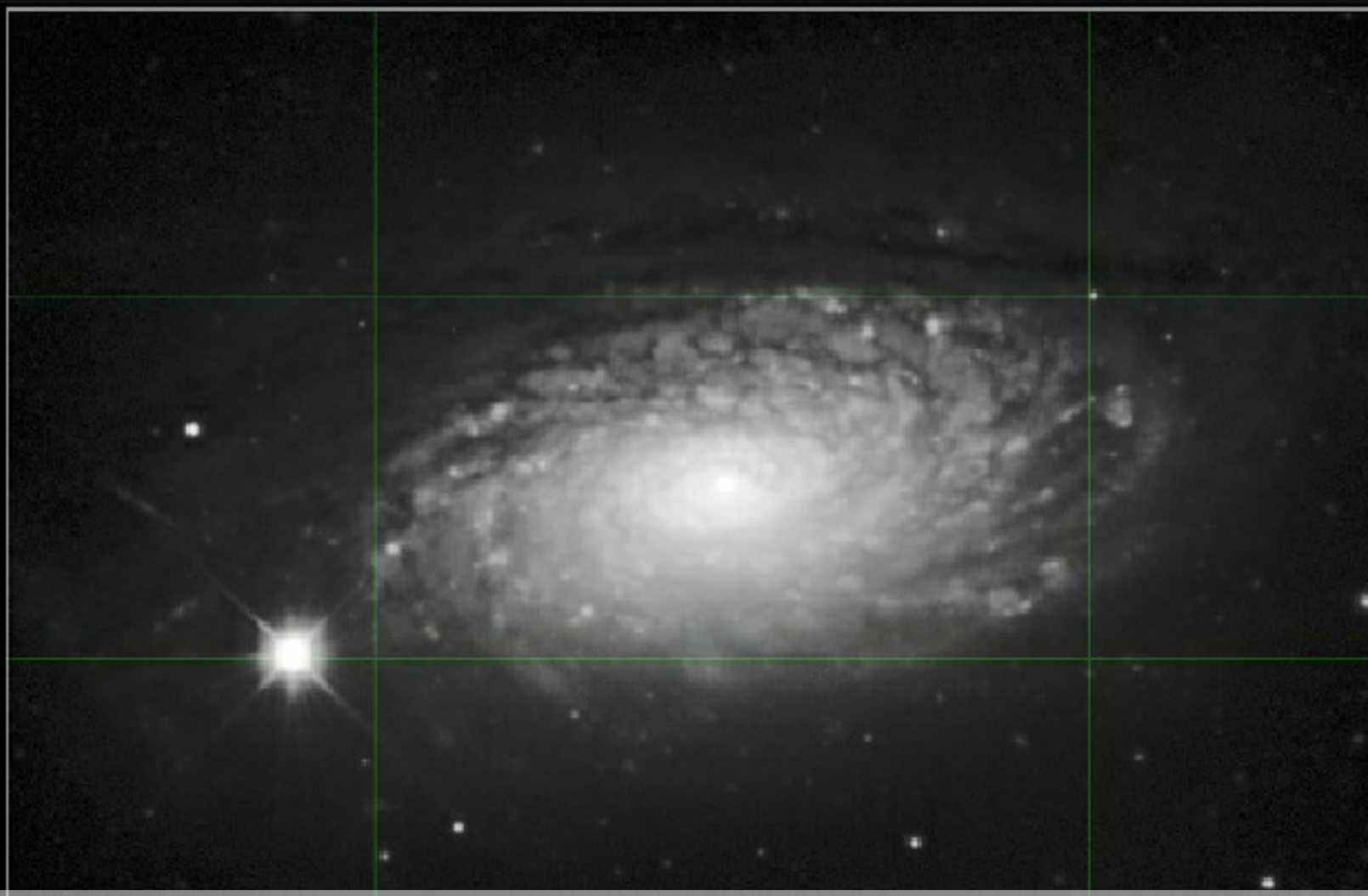
All the RoI needs, is the best possible example of the dynamic range problem it should be solving for. Therefore, you should always give an example that has the widest dynamic range (e.g. has features that run from most dark to most bright). For example, when using AutoDev for the M81 / M82 galaxy pair, it is recommended you choose M81 (a brighter magnitude 6.9) as your RoI and *not* M82 (with a dimmer magnitude of 8.4).

In the above example, should you use M82 rather than M81 as a reference for the RoI, then you will notice M81's core brightening a lot and any detail contained therein being much harder to see. Of course, under no circumstances will the M81 core over-expose completely; a minute amount of dynamic range will always be allocated to it thanks to the '**Outside RoI**' Influence parameter (possibly unless set to 0).

Keeping in mind AutoDev's purpose

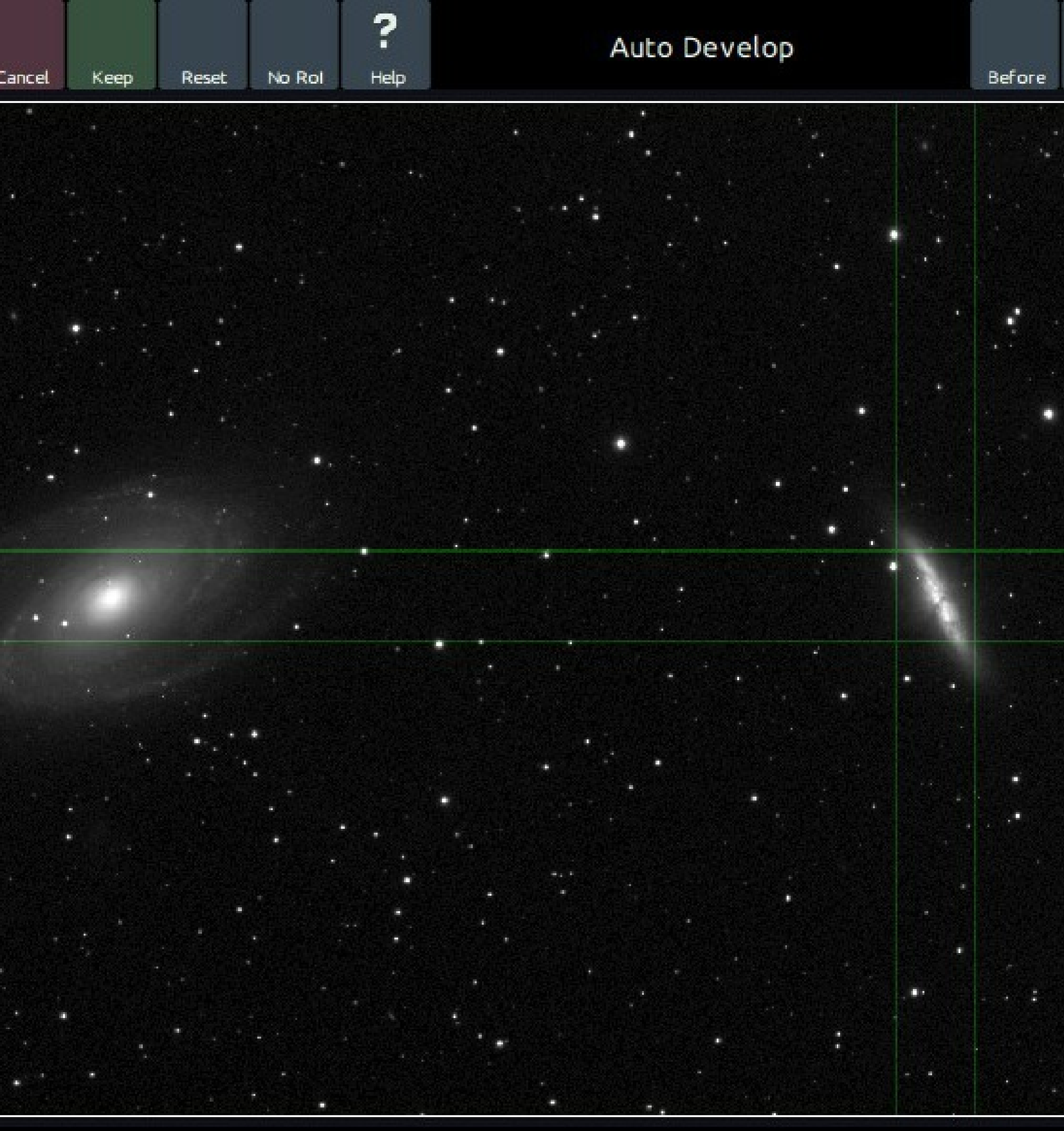
The purpose of AutoDev is to give you the most optimal *global* starting point, ready for enhancement and refinement with modules on a more *local* level. Always keep in the back of your mind that you can use *local* detail restoration modules such as the Contrast, HDR and Sharp modules to *locally* bring out detail. Astrophotography deals with enormous differences in brightness; many objects are their own light source and can range from incredibly bright to incredibly dim. Most astrophotographers strive to show as many interesting astronomical details as possible. StarTools offers you various tools that put you in absolute, objective control over managing these enormous differences in brightness, to the benefit of your viewers.





Confining the Region of Interest ("RoI") progressively to the core of this galaxy, the stretch becomes more and more optimised for th core and less and less for the outer regions.

When you have two objects of interest in your image, choose the one that exhibits the widest dynamic range continuum (magnitude 6.9 M81 in this case).




Choosing an object that is not representative of the dynamic range continuum of all objects, will result in some objects having their shadows or highlights "squashed" (as can be seen in M81) as they are "not of interest", and therefore will only be allocated a very sma amount of dynamic range.

## Color retention

Please note **you should completely disregard the colouring in AutoDev** (if coloring is even at all visible).

Non-linearly stretching an image's RGB components causes its hue and saturation to be similarly stretched and squashed. This is often observable as "washing out" of colouring in the highlights.

Traditionally, image processing software for astrophotography has struggled with this, resorting to kludges like "special" stretching functions (e.g. ArcSinH) that somewhat minimize the problem, or even procedures that make desaturated highlights adopt the colours of neighbouring, non-desaturated pixels.

While other software continues to struggle with colour retention, StarTools Tracking feature allows the Color module to go back in time and completely reconstruct the RGB ratios as recorded, *regardless* of how the image was stretched.

This is one of the major reasons why running the Color module is preferably run as one of the last steps in your processing flow; it is able to completely negate the effect of any stretching - whether global or local - may have had on the hue and saturation of the image.

Because of this, AutoDev's performance is not stymied like some other stretching solutions (e.g. ArcSinH) by a need to preserve colouring. The

two aspects - colour and luminance - of your image are neatly separated thanks to StarTools' signal evolution Tracking engine.

# Bin: Trade Resolution for Noise Reduction

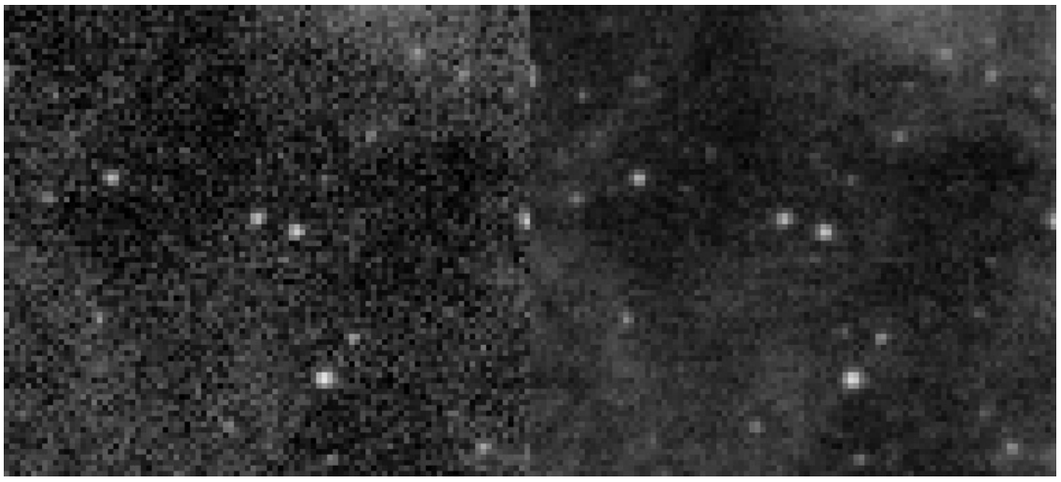
The Bin module puts you in control over the trade-off between resolution, resolved detail and noise.

With today's multi-megapixel imaging equipment and high density CCDs, oversampling is a common occurrence; there is only so much detail that seeing conditions allow for with a given setup. Beyond that it is impossible to pick up fine detail. Once detail no longer fits in a single pixel, but instead gets "smeared out" over multiple pixels due to atmospheric conditions (resulting in a blur), binning may turn this otherwise useless blur into noise reduction. Binning your data may make an otherwise noisy and unusable data set usable again, at the expense of 'useless' resolution.

The Bin module was created to provide a freely scalable alternative to the fixed 2×2 (4x reduction in resolution) or 4×4 (16x reduction in resolution) software binning modes commonly found in other software packages or modern consumer digital cameras and DSLRs (also known as 'Low Light Mode'). As opposed to these other binning solutions, the StarTools' Bin module allows you to bin your data (and gain noise reduction) by the amount you want – if your data is seeing-limited (blurred due to adverse seeing conditions) you are now free to bin your

data until exactly that limit and you are not forced by a fixed 2×2 or 4×4 mode to go beyond that.

Similarly, deconvolution (and subsequent recovery of detail that was lost due to atmospheric conditions) may not be a viable proposition due to the noisiness of an initial image. Binning may make deconvolution an option again. The StarTools Bin module allows you to determine the ratio whith which you use your oversampled data for binning and deconvolution to achieve a result that is finely tuned to your data and imaging circumstances of the night(s).



400% zoomed crop of an image. Left: scaled down to 25% of its original size using nearest neighbor sampling (retaining noise). Right: same image binned down to 25% of its original size. A significant amount of noise reduction has ocurred. Further deconvolution is now an option. Notice real structural detail is not compromised, but any non-structural detail (noise) has been removed.

Core to StarTools' fractional binning algorithm is a custom built anti-

aliasing filter that has been carefully designed to not introduce any ringing (overshoot) and, hence, to not introduce any artefacts when subsequent deconvolution is used on the binned data.

# Usage

The Bin module is operated with just a single parameter; the '**Scale**' parameter. This parameter controls the amount of binning that is performed on the data. The new resolution is displayed ('New Image Size X x Y') , as well the single axis scale reduction, the Signal-to-Noise-Ratio improvement and the increased bit-depth of the new image.

## When to bin?

Data binning is a data pre-processing technique used to reduce the effects of minor observation errors. Many astrophotographers are familiar with the virtues of **hardware** binning. The latter pools the value of 4 (or more) CCD pixels before the final value is read. Because reading introduces noise by itself, pooling the value of 4 or more pixels reduces this 'read noise' also by a factor of 4 (one read is now sufficient, instead of having to do 4). Ofcourse, by pooling 4 pixels, the final resolution is also reduced by a factor of 4. There are many, many factors that influence hardware binning and Steve Cannistra has done a wonderful write-up on the subject on his starrywonders.com website. It also appears that the merits of hardware binning are heavily dependent on the instrument and the chip used.

Most OSCs (One-Shot-Color) and DSLR do not offer any sort of hardware binning in color, due to the presence of a Bayer matrix; binning adjacent pixels makes no sense, as they alternate in the color that they pick up. The best we can do in that case is create a grayscale blend out of them. So hardware binning is out of the question for these instruments.

So why does StarTools offer software binning? Firstly, because it allows us to trade resolution for noise reduction. By grouping multiple pixels into 1, a more accurate 'super pixel' is created that pools multiple measurements into one. Note that we are actually free to use any statistical reduction method that we want. Take for example this 2 by 2 patch of pixels;

7 7
3 7

A 'super pixel' that uses simple averaging yields (7 + 7 + 3 + 7) / 4 = 6. If we suppose the '3' is anomalous value due to noise and '7' is correct, then we can see here how the other 3 readings 'pull up' the average value to 6; pretty darn close to 7.

We could use a different statistical reduction method (for example taking the median of the 4 values) which would yield 7, etc. The important thing is that grouping values like this tends to filter out outliers and make your super pixel value more precise.

## Binning and the loss of resolution

But what about the downside of losing resolution? That super high resolution *may* have actually been going to waste! If for example your CCD can resolve detail at 0.5 arcsecs per pixel, but your seeing is at best 2.0 arcsecs, then you effectively have 4 times more pixels than you need to record one 1 unit of real resolvable celestial detail. Your image will be "oversampled", meaning that you have allocated more resolution than the signal really will ever require. When that happens, you can zoom in into your data and you will notice that all fine detail looks blurry and smeared out over multiple pixels. And with the latest DSLRS having sensors that count 20 million pixels and up, you can bet that most of this resolution will be going to waste at even the most moderate magnification. Sensor resolution may be going up, but the atmosphere's resolution will forever remain the same - buying a higher resolution instrument will do nothing for the detail in your data in that case! This is also the reason why professional CCDs are typically much lower in resolution; the manufacturers rather use the surface area of the chip for coarser but more deeper, more precise CDD wells ('pixels') than squeezing in a lot of very imprecise (noisy) CCD wells (it has to be said the latter is a slight oversimplification of the various factors that determine photon collection, but it tends to hold).

## Binning to undo the effects of debayering interpolation

There is one other reason to bin OSC and DSLR data to at least 25% of its original resolution; the presence of a bayer matrix means that (assuming an RGGB matrix) after applying a debayering (aka 'demosaicing') algorithm, 75% of all red pixels, 50% of all green pixels, and another 75% of all blue pixels are completely made up!

Granted, your 16MP camera may have a native resolution of 16 million pixels, however it has to divide these 16 million pixels up between the red, green and blue channels! Here is another very good reason why you might not want to keep your image at native resolution. Binning to 25% of native resolution will ensure that each pixel corresponds to one real recorded pixel in the red channel, one real recorded pixel in the blue channel and two pixels in the green channel (the latter yielding a 50% noise reduction in the green channel).

There are, however, instances where the interpolation can be undone if enough frames are available (through sub-pixel dithering) to have exposed all sub-pixels of the bayer matrix to real data in the scene (drizzling).

## Fractional binning

StarTools' binning algorithm is a bit special in that it allows you to apply 'fractional' binning; you're not stuck with pre-determined factors (ex. 2x2, 3x3 or 4x4). You can bin exactly the amount that achieves a single unit of celestial detail in a single pixel. In order to see what that limit is, you simply keep reducing resolution until no blurriness can be detected when zooming into the image. Fine detail (not noise!) should look crisp. However, you may decide to leave a little bit of blurriness to see if you can bring out more detail using deconvolution.

Operating the Bin module is easy, with just one slider doing all the work.

# Color: Advanced Color Correction and Manipulation

Thanks to StarTools' Tracking feature the Color module provides you with unparalleled flexibility and color fidelity when it comes to colour presentation in your image.

The Color module fully capitalises on the signal processing engine's unique ability to process chrominance and detail separately, yet simulatenously. This unique capability is responsible for a number of innovative features.

Firstly, whereas other software without Tracking data mining, destroys colour and colour saturation in bright parts of the image as the data gets stretched, StarTools allows you to retain colour and saturation throughout the image with its 'Color Constancy' feature. This ability allows you to display all colours in the scene as if it were evenly illuminated, meaning that even very bright cores of galaxies and nebulas retain the same colour throughout, irrespective of their local brightness, or indeed acquisition methods and parameters.

This ability is important in scientific representation of your data, as it allows the viewer to compare similar objects or areas like-for-like, since colour in outer space very often correlates with chemical signatures or temperature.

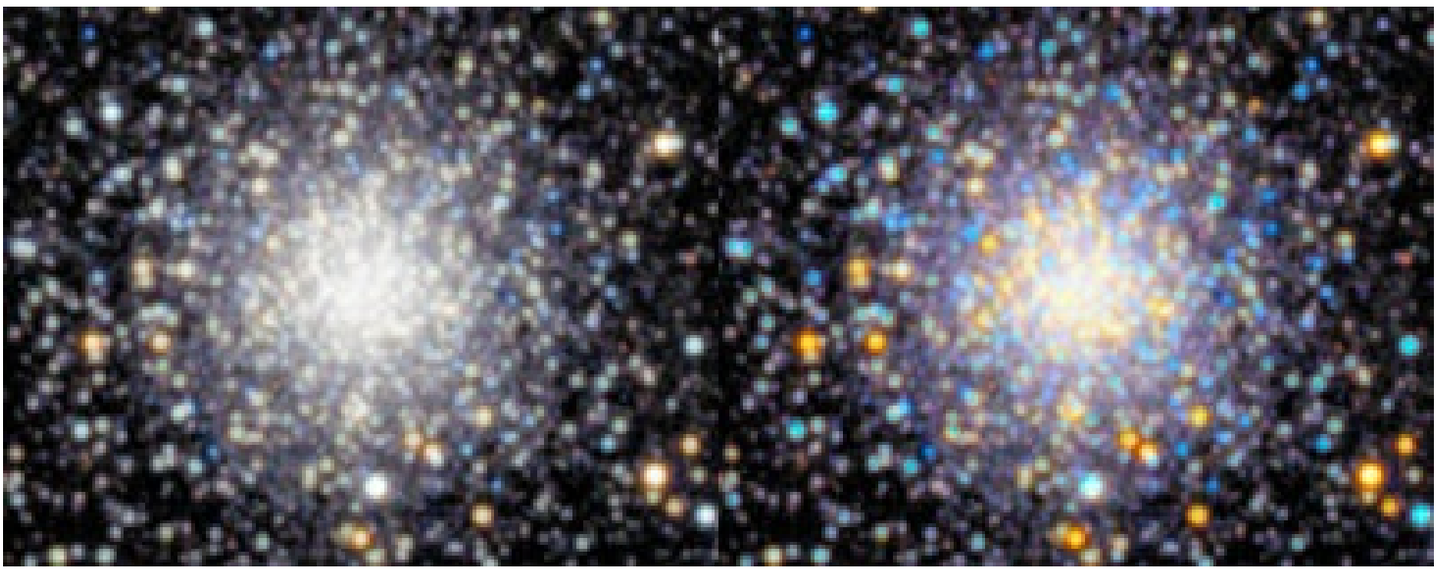
The same is true for star temperatures across the image, even in bright, dense star clusters. This mode allows the viewer of your image to objectively compare different parts and objects in the image without suffering from reduced saturation in bright areas. It allows the viewer to explore the universe that you present in full colour, adding another dimension of detail, irrespective of the exposure time and subsequent stretching of the data.

For example, StarTools enables you to keep M42's colour constant throughout, even in its bright core. No fiddling with different exposure times, masked stretching or saturation curves needed. You are able to show M31's true colours instead of a milky white, or resolve star temperatures to well within a globular cluster's bright core. All that said, if you're a fan of the traditional 'handicapped' way of colour processing in other software, then StarTools can emulate this type of processing as well.

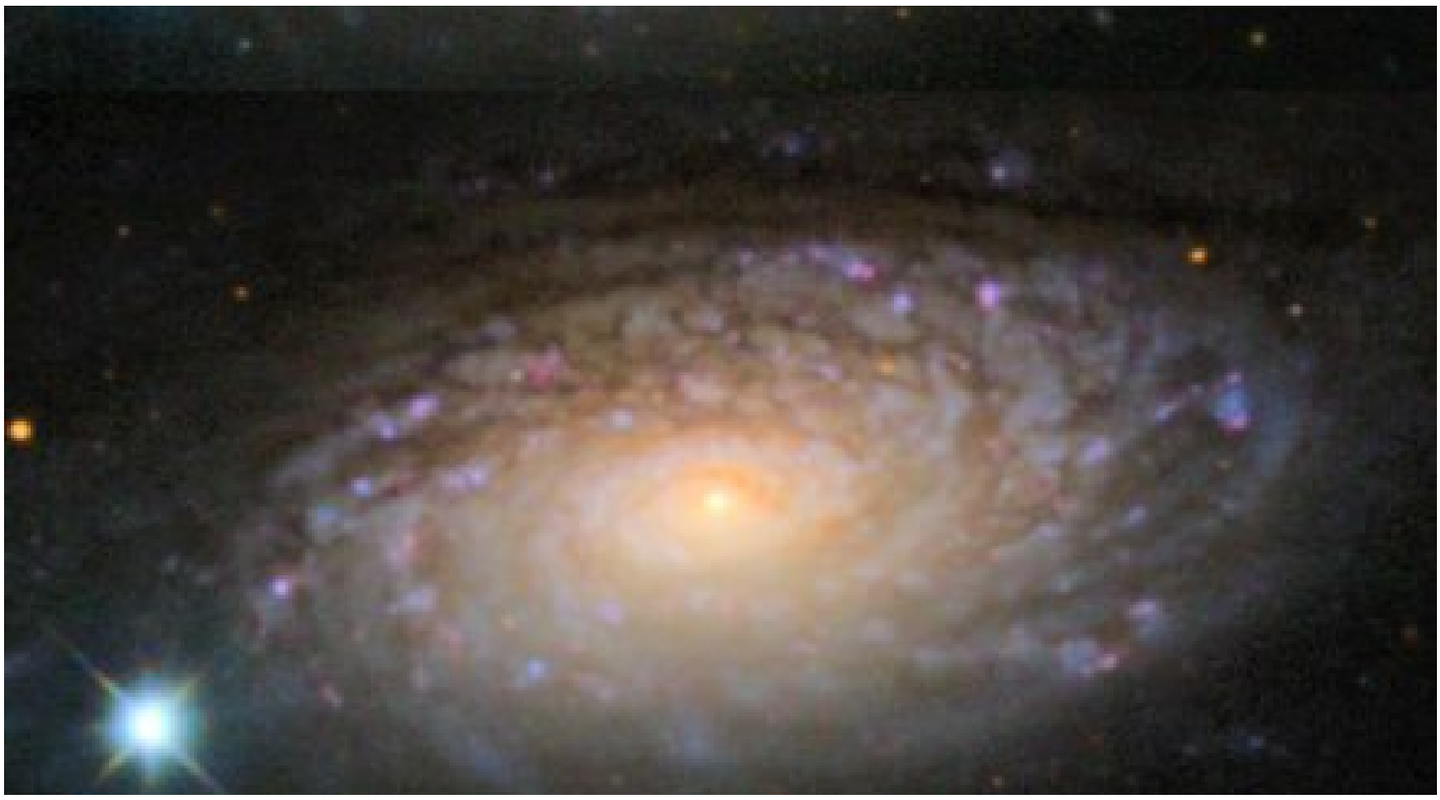
The Color module's abilities don't stop there, however. It is also capable of emulating a range of complex LRGB color compositing methods that have been invented over the years. And it does it at the click of a button. Even if you acquired data with an OSC or DSLR, you will still be able to use these compositing methods; the Color module will generate synthetic luminance from your RGB on the fly and re-composite the image in your desired compositing style.

The Color module allows for various ways to calibrate the image, including by star field, galaxy sampling and - unique to StarTools - the MaxRGB calibration view. The latter allows for objective colour calibration, even on poorly calibrated screens.

Because luminance (detail) and chrominance is processed separately in parallel, the module is capable of remapping channels *for the purpose of colour* (aka "tone mapping") on the fly, without impacting detail. The result is the unique ability to flip between popular colour renditions for, for example, narrowband data with a single click, whether you are processing SHO/HST datasets or duo/tri/quadband datasets. Similarly, DSLR users benefit from the ability to use the manufacturer's preferred colour matrix, yet without the cross-channel noise contamination that would otherwise impact luminance (detail).



Left: traditional processing, Right: StarTools color constancy showing star temperatures evenly until well into the core.



Top: traditional processing, Bottom: StarTools color constancy showing true color of the core, regardless of brightness. (image acquisition by Jim Misti)

Color constancy (right) demonstrates how features with similar chemical/phsycial properties show identical colors, regardless of brightness.

# Usage

The Color module is very powerful - offering capabilities surpassing most other software - yet it is simple to use.

The primary goal that the Color module was designed to accomplish, is achieving a good colour balance that accurately describes the colour ratios that were recorded. In accomplishing that goal, the Color module goes further than other software by offering a way to negate the adverse effects of non-linear dynamic range manipulations on the data (thanks to Tracking data mining). In simple terms, this means that colouring can be reproduced (and compared!) in a consistent manner regardless of how bright or dim a part of the scene is shown.

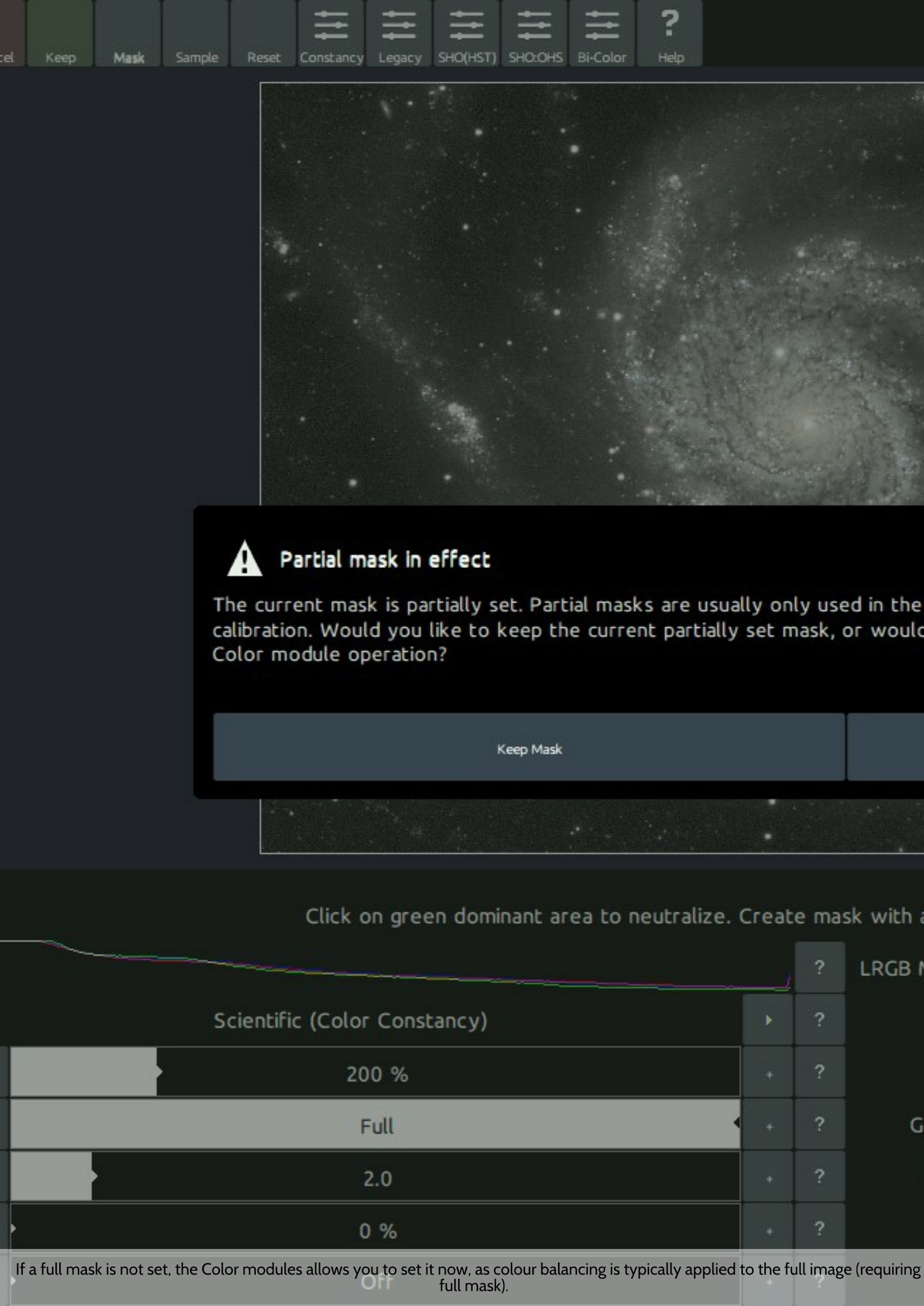
A second unique feature of StarTools, is its ability to process luminance (detail) and chrominance (colour) separately, yet simultaneously. This means that any decisions you make affecting your detail does not affect the colouring of said detail, and vice-versa. This ability further allows you to remap colour channels (aka "tone mapping") for narrowband data, without having to start over with your detail processing. This lets you try out many different popular color schemes at the click of a button.

## Launching the Color module

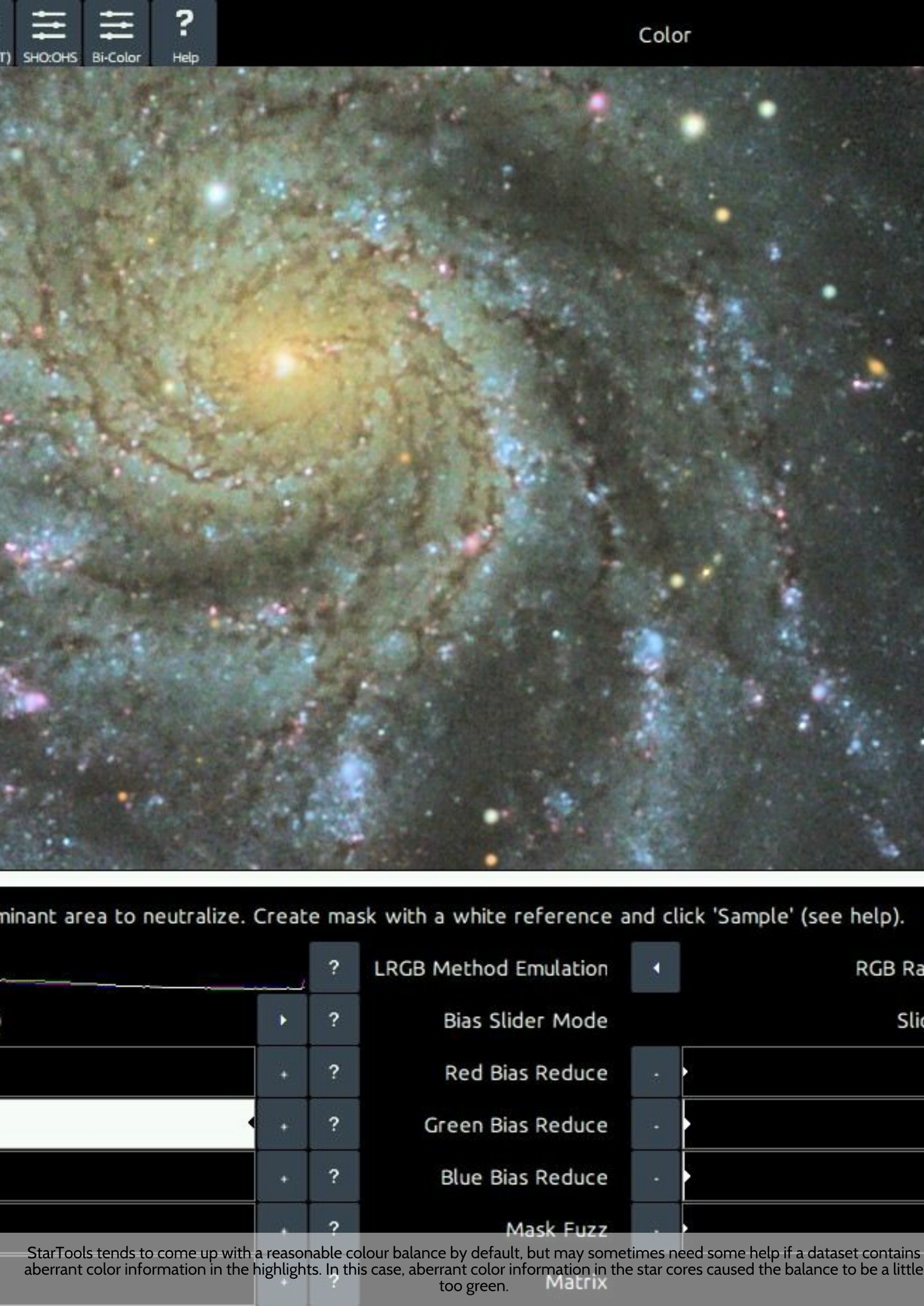
Upon launch, the colour module blinks the mask three times in the familiar way. If a full mask is not set, the Color modules allows you to set it now, as colour balancing is typically applied to the full image (requiring a full mask).

In addition to blinking the mask, the Color module also analyses the image and sets the **'Red, Green and Blue Increase/Reduce'** parameters to a value which it deems the most appropriate for your image. This behaviour is identical to manually clicking the 'Sample' button where the whole image is sampled.

In cases where the image contains aberrant colour information in the highlights, for example due to chromatic aberration or slight channel misalignment/discrepancies, then this initial colour balance may be significantly incorrect and may need further correction. The aberrant colour information in the highlights itself, can be repaired using the **'Highlight Repair'** parameter.




If a full mask is not set, the Color modules allows you to set it now, as colour balancing is typically applied to the full image (requiring full mask).




StarTools tends to come up with a reasonable colour balance by default, but may sometimes need some help if a dataset contains aberrant color information in the highlights. In this case, aberrant color information in the star cores caused the balance to be a little too green.

Aberrant color information due to misalignments/fringing or chromatic aberration, may throw off the initial colour balance.

## Setting a colour balance

The **'Red, Green and Blue Increase/Reduce'** parameters are the most important settings in the Color module. They directly determine the colour balance in your image. Their operation is intuitive; is there too much red in your image? Then increase the **'Red Bias Reduce'** value. Too little red in your image? Reduce the **'Red Bias Reduce'** value.

If you would rather operate on these values in terms of Bias *Increase*, then simply switch the **'Bias Slider Mode'** setting to 'Sliders Increase Color Bias'. The values are now represented in terms of relative increases, rather than decreases. Switching between these two modes you can see that, for example, a Red Bias Reduce of 8.00 is the same as a Green *and* Blue Bias *Increase* of 8.00. This should make intuitive sense; a relative decrease of red makes blue and green more prevalent and vice versa.

The Red, Green and Blue Bias controls.

## Color balancing techniques

Now that we know how to change the colour balance, how do we know what to actually set it to?

The goal of color balancing in astrophotography, is achieving an accurate representation of emissions, temperatures and processes. A visual spectrum dataset should show emissions where they occur in the blend of colours they occur in. A narrowband dataset, equally, should be rendered as an accurate representation of *the relative ratio* of emissions (but not necessarily with the color they correspond to the wavelength they appear at in the visual spectrum). So, in all cases, whether your dataset is a visual spectrum dataset or a narrowband dataset, it should let your viewers allow to compare different areas in your image and accurately determine what emissions are dominant, where.

There are a great number of tools and techniques that can be applied in StarTools that let you home in on a good colour balance. Before delving into them, It is highly recommended to switch the '**Style**' parameter to 'Scientific (Color Constancy)' during colour balancing, even if that is not the preferred style of rendering the colour of the end result, this is because the Color Constancy feature makes it much easier to colour balance by eye in some instances due to its ability to show continuous, constant colour throughout the image. Once a satisfactory colour balance is achieved you should, of course, feel free to switch to any alternative style of colour rendering.

## White point reference by mask sampling

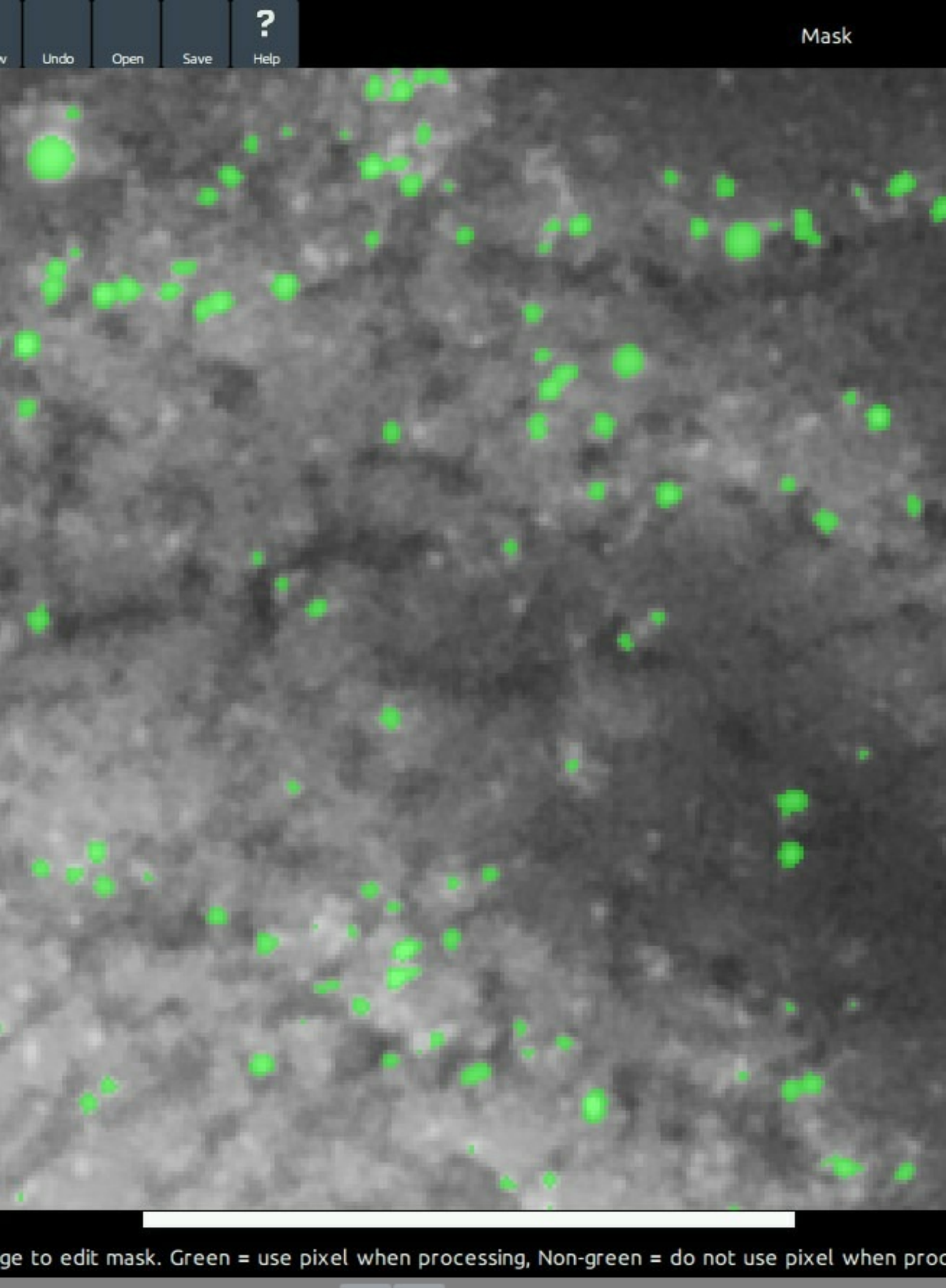
Upon launch the Color module samples whatever mask is set (note also that the set mask also ensures the Color module only applies any changes to the masked-in pixels!) and sets the **'Red, Green and Blue Increase/Reduce'** parameters accordingly.

We can use this same behaviour to sample larger parts of the image that we know should be white. This method mostly exploits the fact that stars come in all sorts of sizes and temperatures (and thus colours!) and that this distribution is usually completely random in a wide enough field. Indeed, the Milky Way is named as such because the average color of all its stars is perceived as a milky white. Therefore if we sample a large enough population of stars, we should find the average star color to be - likewise - white .

We can accomplish that in two ways; we either sample all stars (but only stars!) in a wide enough field, or we sample a whole galaxy that happens to be in the image (note that the galaxy must be of a certain type to be a good candidate and be reasonably close - preferably a barred spiral galaxy much like our own Milkyway).

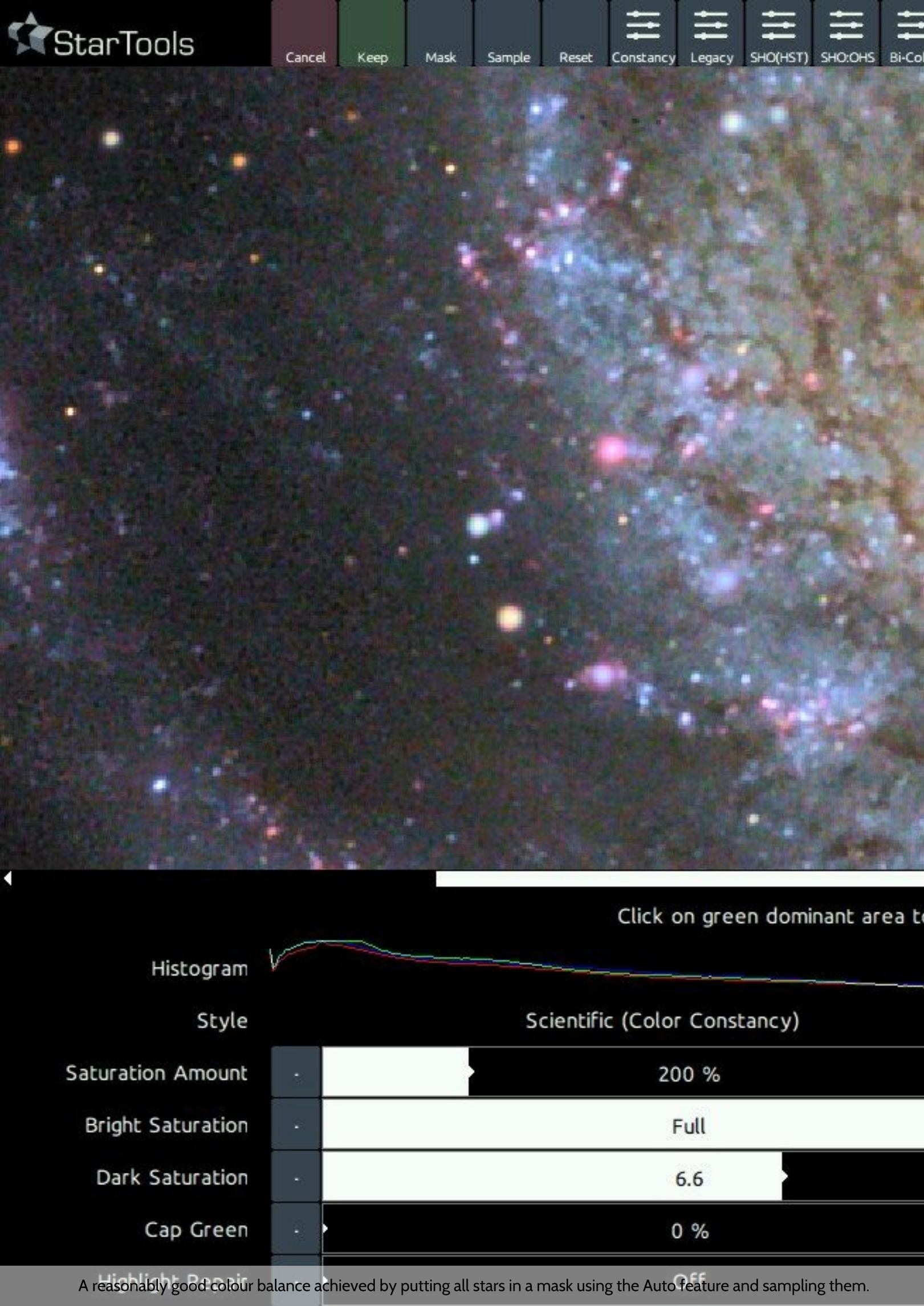
Whichever you choose, we need to create a mask, so we launch the Mask editor. Here we can use the Auto feature to select a suitable selection of stars, or we can us the Flood Fill Brighter or Lassoo tool to select a galaxy. Once selected, return to the Color module and click Sample. StarTools will now determine the correct **'Red, Green and Blue**

**Increase/Reduce'** parameters to match the white reference pixels in the mask so that they come out neutral.




We can calibrate against a big enough population of non-associated foreground stars, by putting them in a mask, clicking 'Sample' in the Color module and applying the found bias values to the whole image again.

To apply the new colour balance to the whole image, launch the Mask editor once more and click Clear, then click Invert to select the whole image. Upon return to the Color module, the whole image will now be balanced by the Red, Green and Blue bias values we determined earlier with just the white reference pixels selected.




A reasonably good colour balance achieved by putting all stars in a mask using the Auto feature and sampling them.

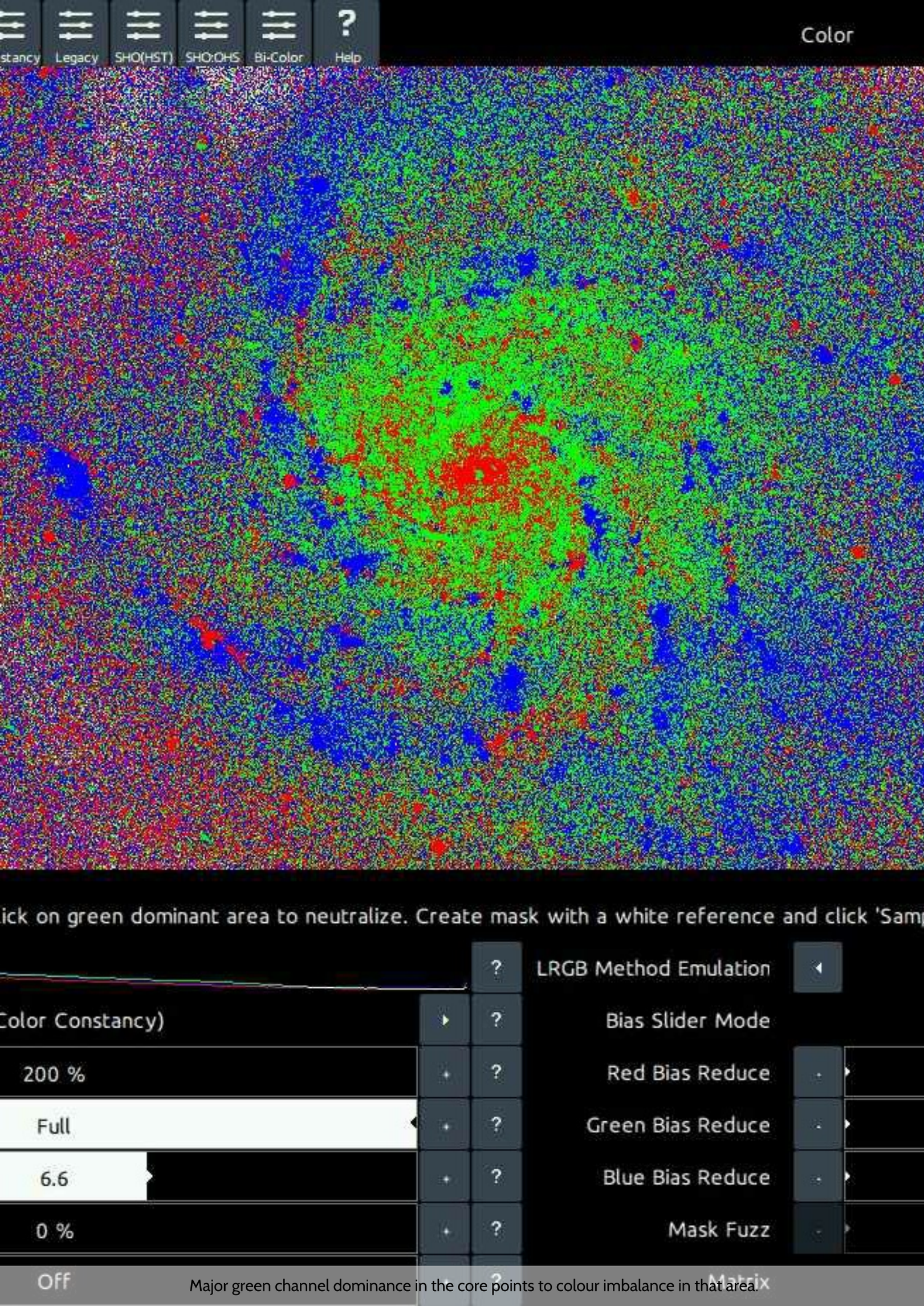
## White balancing in MaxRGB mode

StarTools comes with a unique colour balancing aid called MaxRGB. This mode of colour balancing is exceptionally useful if trying to colour balance by eye, but the user suffers from colour blindness or uses a screen that is not colour calibrated very well. The mode can be switched on or off by clicking on the MaxRGB mode button in the top right corner.

The MaxRGB aid allows you to view which channel is dominant per-pixel. If a pixel is mostly red, that pixel is shown red, if a pixel is mostly green, that pixel is shown green, and if a pixel is mostly blue, that pixel is shown blue.

By cross referencing the normal image with the MaxRGB image, it is possible to find deficiencies in the colour balance. For example, the colour green is very rarely dominant in space (with the exception of highly dominant OIII emission areas in, for example the Trapezium in M42).

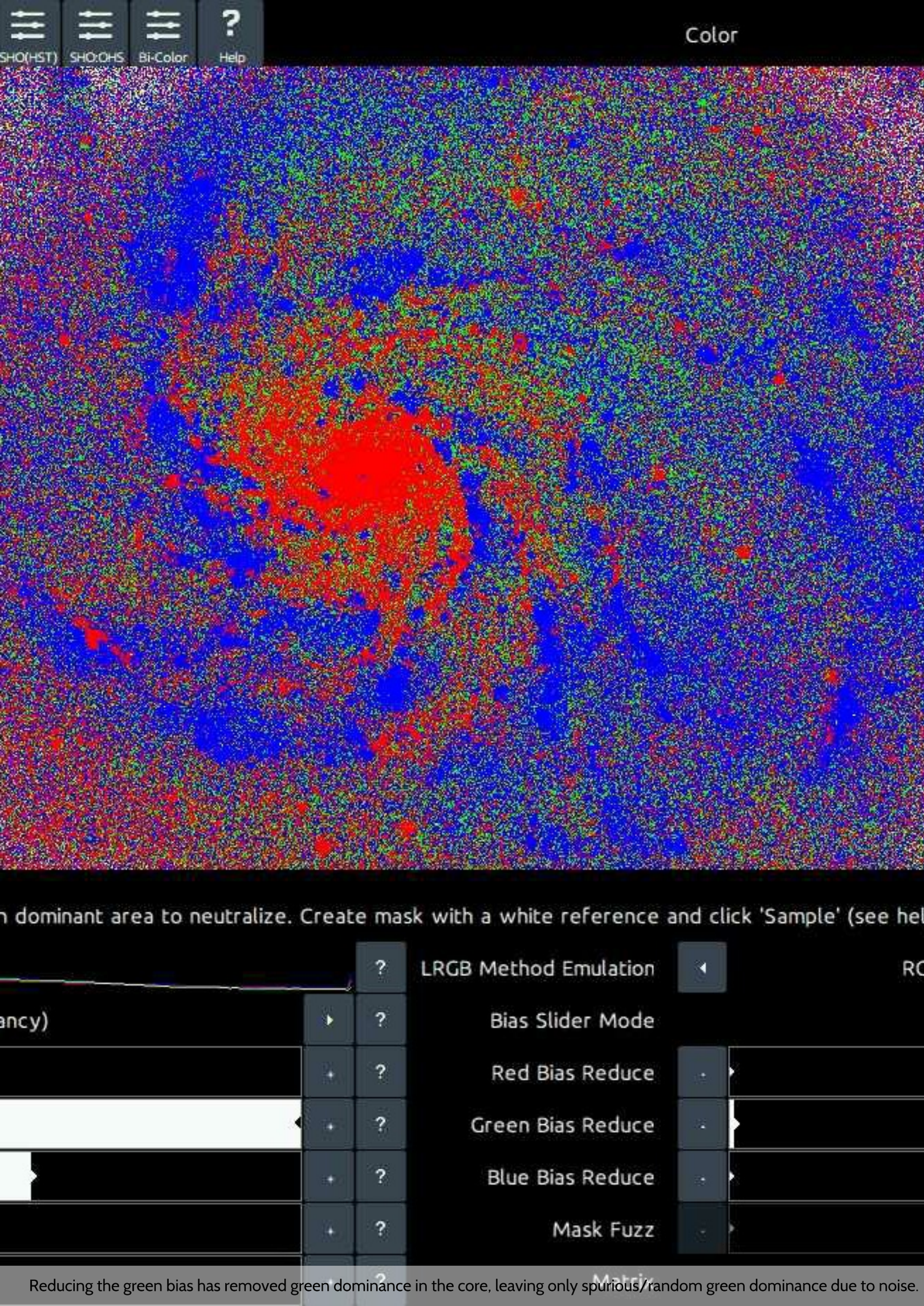
Therefore, if we see large areas of green, we know that we have too much green in our image and we should adjust the bias accordingly. Similarly if we have too much red or blue in our image, the MaxRGB mode will show many more red than blue pixels in areas that should show an even amount (for example the background). Again we then know we should adjust red or green accordingly.



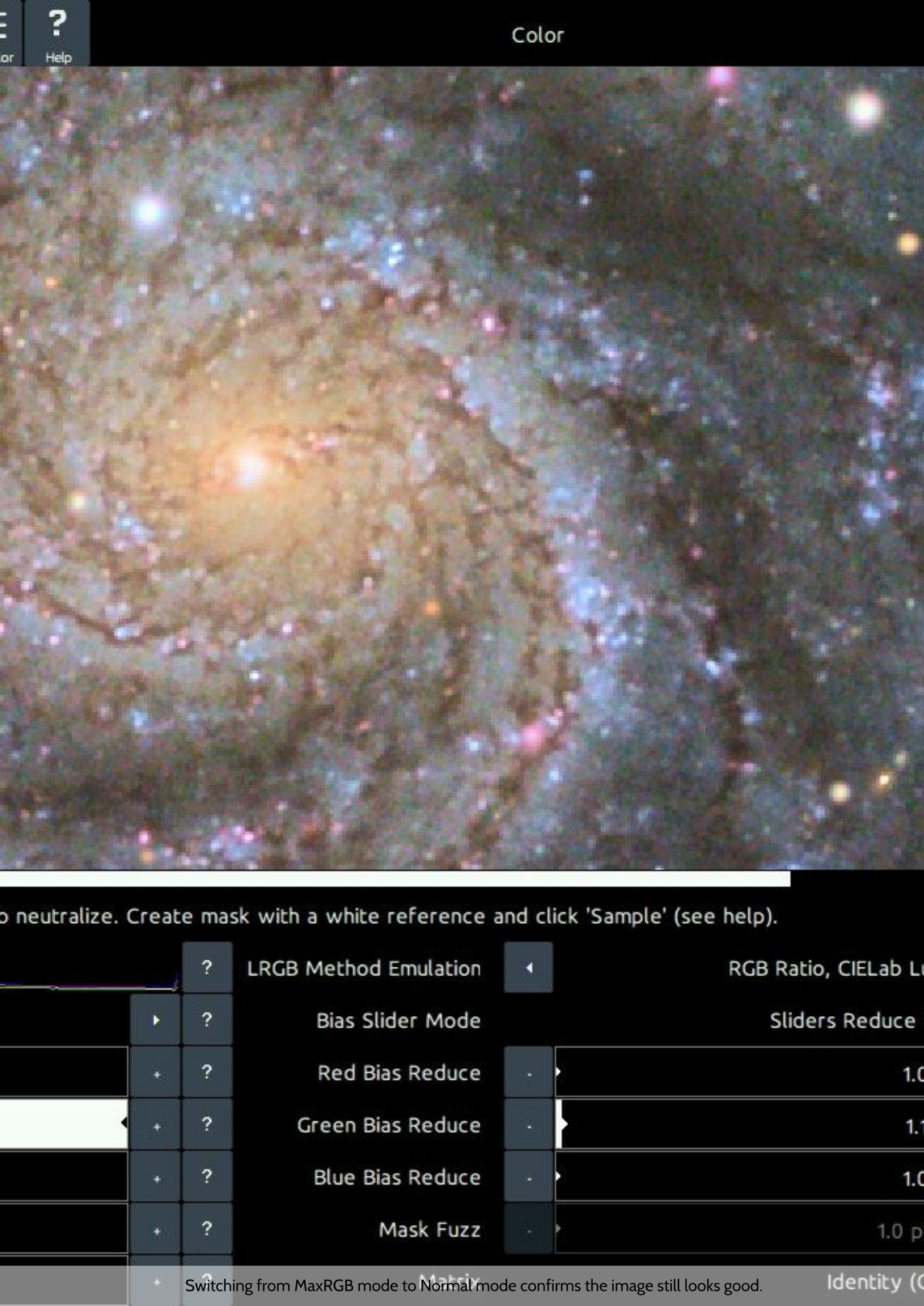

Major green channel dominance in the core points to colour imbalance in that area.

Clicking on an area to neutralise green

A convenient way to eliminate green dominance is to simply click on an area. The Color module with adjust the '**Green Bias Reduce**' or '**Green Bias Increase**' in response so that any green dominance in that area is neutralised.




Reducing the green bias has removed green dominance in the core, leaving only spurious/random green dominance due to noise.




Switching from MaxRGB mode to Normal mode confirms the image still looks good.

## White balancing by known features and processes

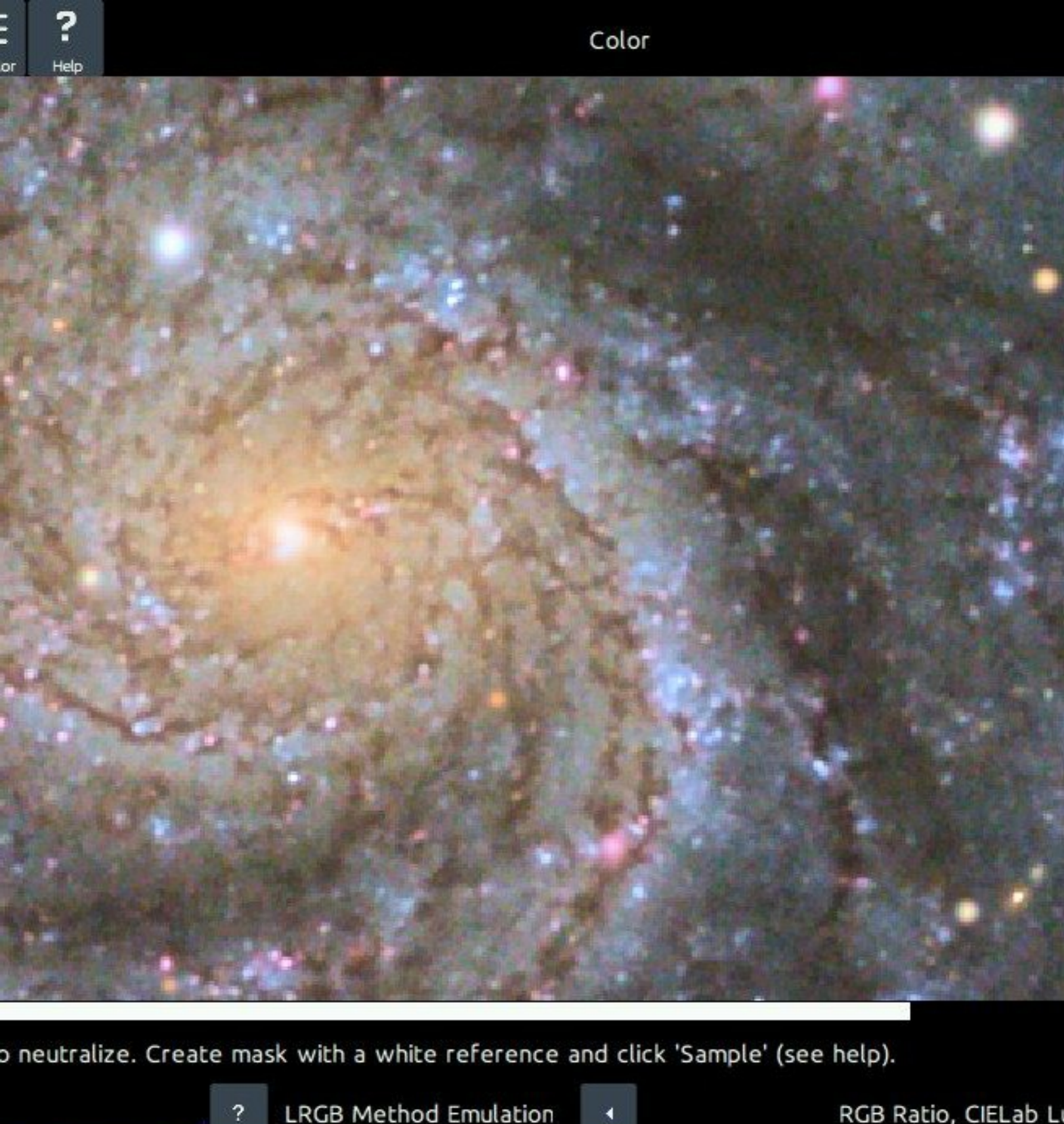
StarTools' Color Constancy feature makes it much easier to see colours and spot processes, interactions, emissions and chemical composition in objects. In fact, the Color Constancy feature makes colouring comparable between different exposure lengths and different gear. This allows for the user to start spotting colours repeating in different features of comparable objects. Such features are, for example, the yellow cores of galaxies (due to the relative over representation of older stars as a result of gas depletion), the bluer outer rims of galaxies (due to the relative over representation of bright blue young stars as a result of the abundance of gas) and the pink/purplish HII area 'blobs' in their discs. Red/brown (white light filtered by dust) dust lanes complement a typical galaxy's rendering.

Similarly, HII areas in our own galaxy (e.g. most nebulae), while in StarTools Color Constancy Style mode, display the exact same colour signature found in the galaxies; a pink/purple as a result of predominantly deep red Hydrogen-alpha emissions mixed with much weaker blue/green
emissions of Hydrogen-beta and Oxygen-III emissions and (more dominantly) reflected blue star light from bright young blue giants who are often born in these areas, and shape the gas around them.

Dusty areas where the bright blue giants have 'boiled away' the Hydrogen through radiation pressure (for example the Pleiades) reflect

the blue star light of any surviving stars, becoming distinctly blue reflection nebulae. Sometimes gradients can be spotted where (gas-rich) purple gives away to (gas-poor) blue (for example the Rosette core) as this process is caught in the act.

Diffraction spikes, while artefacts, also can be of great help when calibrating colours; the "rainbow" patterns (though skewed by the dominant colour of the star whose light is being diffracted) should show a nice continuum of colouring.




M101 exhibiting a nice yellow core, bluer outer regions, red/brown dust lanes and purple HII knots, while the foreground stars show a good distribution of color temperatures from red to orange, yellow, white to blue.

Finally, star temperatures, in a wide enough field, should be evenly distributed; the amount of red, orange, yellow, white and blue stars should be roughly equal. If any of these colors are missing or are over-represented we know the colour balance is off.

## Colour balancing of data that was filtered by a light pollution filter

Colour balancing of data that was filtered by a light pollution filter is fundamentally impossible; narrow (or wider) bands of the spectrum are missing and no amount of colour balancing is going to bring them back and achieve proper colouring. A typical filtered data set will show a distinct lack in yellow and some green when properly colour balanced. It's by no means the end of the world - it's just something to be mindful of.

Correct colouring may be achieved however by shooting deep luminance data with light pollution filter in place, while shooting colour data without filter in place, after which both are processed separately and finally combined. Colour data is much more forgiving in terms of quality of signal and noise; the human eye is much more sensitive to noise in the luminance data that it is in the colour data. By making clever use of that fact and performing some trivial light pollution removal in Wipe, the best of both worlds can be achieved.

A visual spectrum colour balance will not be possible with datasets shot through a light pollution filter, however pleasing results showing important coloring (for example emissions and reflection nebulosity) quite accurately, can still be achieved.

## OSC (One-Shot-Color) instruments

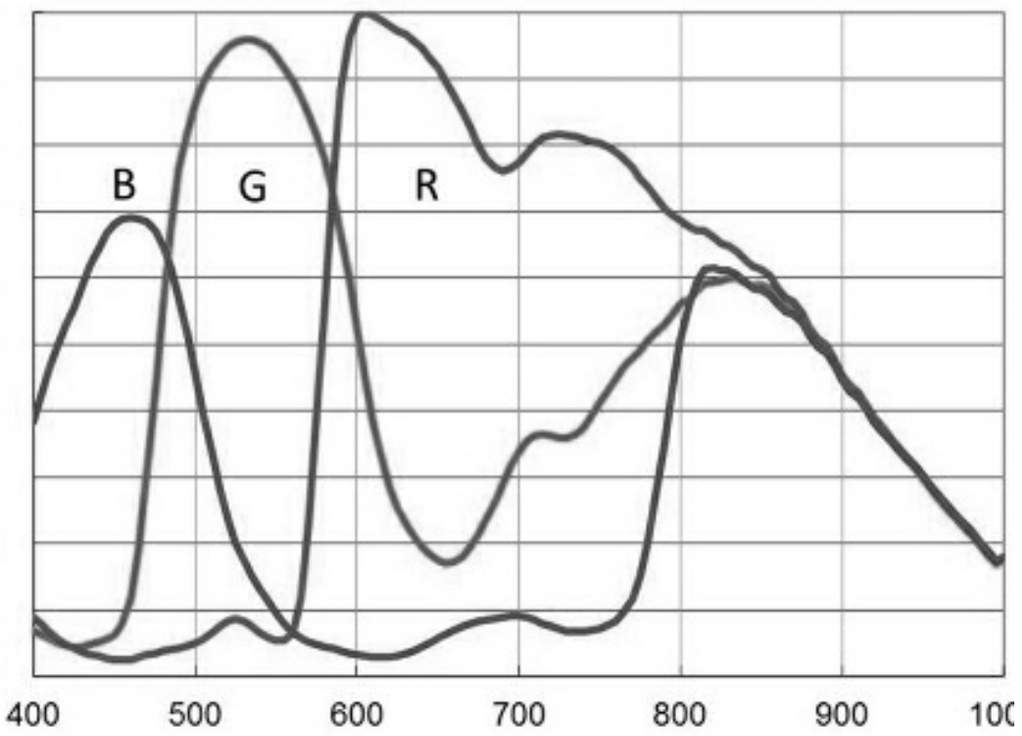
Many modern OSC cameras have a spectrum response that increases in sensitivity *across all channels* beyond the visual spectrum red cut-off (the human eye can detect red wavelengths up until around 700nm). This is a feature that allows these cameras pick up detail beyond the visual spectrum (for example for use with narrowband filters or for recording infrared detail).

However, imaging with these instruments without a suitable IR/UV filter (also known as a "luminance filter") in place, will cause these extra-visual spectrum wavelengths to accumulate in the visual spectrum channels. This can significantly impact the "correct" (in terms of visual spectrum) colouring of your image. Just as a light pollution filter makes it fundamentally impossible to white-balance back the missing signal, so too does imaging with extended spectrum response make it impossible to white-balance the superfluous signal away.

Hallmarks of datasets that have been acquired with such instruments, without a suitable IR/UV filter in place, is a distinct yellow cast that is hard (impossible) to get rid of, due to a strong green response coming back in combined with extended red channel tail.

The solution is to image with a suitable IR/UV filter in place that cuts-off the extended spectrum response before those channels increase in sensitivity again. The needed IR/UV filter will vary per OSC. Consult the respective manufacturers' spectral graphs to find the correct match for

your OSC.




This example spectral response graph of a ZWO ASI290MC camera shows a marked "bump" in the green and blue response beyond the red visual spectrum cut-off (approximately 700nm).

## Tweaking your colors

Once you have achieved a color balance you are happy with, the StarTools Color module offers a great number of ways to change the presentation of your colours.

## Style

The parameter with the biggest impact is the '**Style**' parameter. StarTools is renowned for its Color Constancy feature, rendering colours in objects regardless of how the luminance data was stretched, the reasoning being that colours in outer space don't magically change depending on how we stretch our image. Other software sadly lets the user stretch the colour information along with the luminance information, warping, distorting and destroying hue and saturation in the process. The 'Scientific (Color Constancy)' setting for Style undoes these distortions using Tracking information, arriving at the colours as recorded.

To emulate the way other software renders colours, two other settings are available for the '**Style**' parameter. These settings are ''Artistic, Detail Aware'' and ''Artistic, Not Detail Aware''. The former still uses some Tracking information to better recover colours in areas whose dynamic range was optimised locally, while the latter does not compensate for any distortions whatsoever.

## LRGB Method Emulation

The '**LRGB Method Emulation**' parameter allows you to emulate a number of colour compositing methods that have been invented over the years. Even if you acquired data with an OSC or DSLR, you will still be able to use these compositing methods; the Color module will generate synthetic luminance from your RGB on the fly and re-composite the image in your desired compositing style.

The difference in colouring can be subtle or more pronounced. Much depends on the data and the method chosen.

- 'Straight CIELab Luminance Retention' manipulates all colours in a psychovisually optimal way in CIELab space, introducing colour without affecting apparent 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, using RGB ratios multiplied by luminance in order to better preserve star colour. 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 colour information with a 50% opacity. Luminance retention in CIELab color space is applied afterwards. The inherent loss of 50% in saturation is compensated for, for your convenience, in order to allow 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, using RGB ratios multiplied by luminance in order to better preserve star colour. No further luminance retention is attempted.
- '50/50 Layering, CIELab Luminance Retention' uses a method proposed by Robert Gendler, where luminance is layered on top of

the colour information with a 50% opacity. No further luminance retention is attempted. The inherent loss of 50% in saturation is compensated for, for your convenience, in order to allow for easier comparison with other methods.

When processing a complex composite that carries a luminance signal that is substantially decoupled from the chrominance signal (for example importing H-alpha as luminance and a visual spectrum dataset as red, green and blue via the Compose module), then the 'RGB Ratio, CIELab Luminance Retention' will typically do a superior job accommodating the greater disparities in luminance and how this affect final colouring.

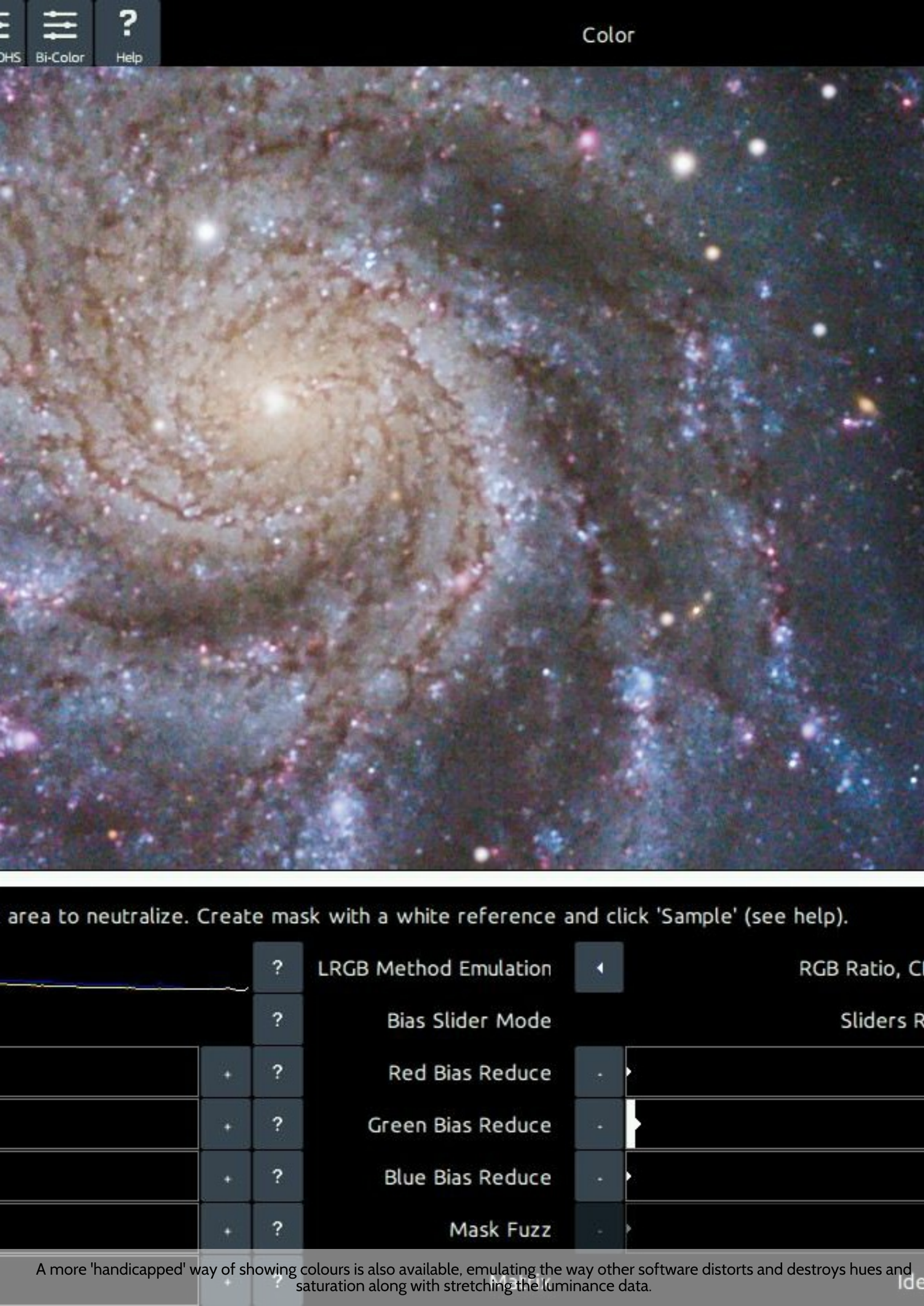
Finally, please note that the LRGB Emulation Method feature is only available when Tracking is engaged.

Saturation

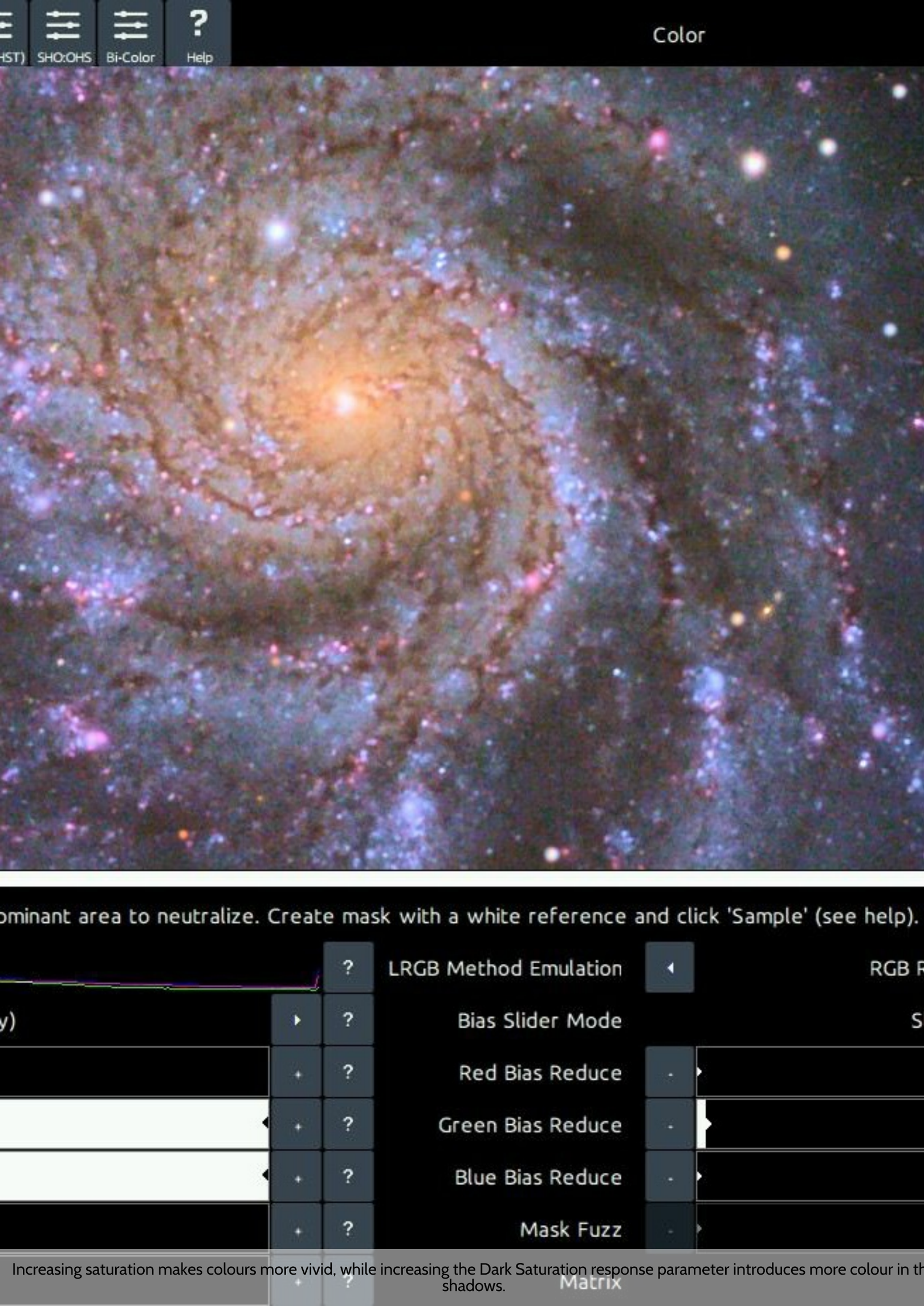
The '**Saturation**' parameter allows colours to be rendered more, or less vividly, whereby the '**Bright Saturation**' parameter and '**Dark Saturation**' parameter control how much colour and saturation is introduced in the highlights and shadows respectively. It is important to note that introducing colour in the shadows may exacerbate colour noise, though Tracking will make sure any such noise exacerbations are recorded and dealt with during the final denoising stage.

## Cap Green

The '**Cap Green**' parameter, finally, removes spurious green pixels if needed, reasoning that green dominant colours in outer space are rare and must therefore be caused by noise. Use of this feature should be considered a last resort if colour balancing does not yield adequate results and the green noise is severe. The final denoising stage should, thanks to Tracking data mining, pin pointed the green channel noise already and should be able to adequately mitigate it.




A more 'handicapped' way of showing colours is also available, emulating the way other software distorts and destroys hues and saturation along with stretching the luminance data.




Increasing saturation makes colours more vivid, while increasing the Dark Saturation response parameter introduces more colour in th shadows.

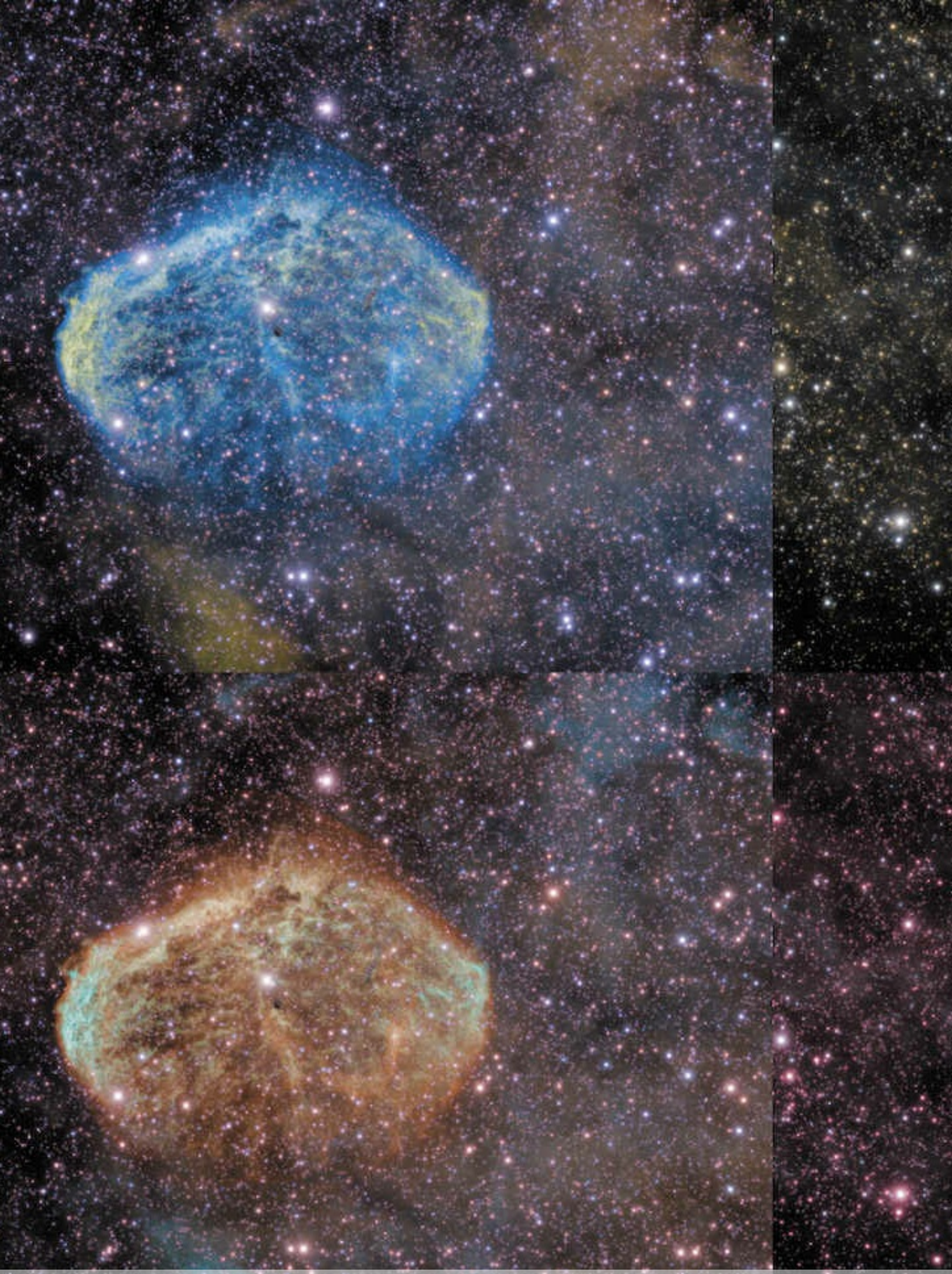
## Matrix correction and on-the-fly channel remapping

The Color module comes with a vast number of camera color correction matrices for various DSLR manufacturers (Canon, Nikon, Sony, Olympus, Pentax and more), as well as a vast number of channel blend remappings (aka "tone mapping") for narrowband dataset (e.g. HST/SHO or bi-color duoband/quadband filter data).

Uniquely, thanks to the signal evolution Tracking engine, color calibration is preferably performed *towards the end* of your processing workflow. This allows you to switch color rendering at the very last moment at the click of a button without having to re-composite and re-process, while also allowing you to use cleaner, non-whitebalanced, non-matrix corrected data for your luminance component, aiding signal fidelity.

Camera Matrix correction is performed *towards the end* of your processing workflow on your chrominance data only, rather than in the RAW converter during stacking. This helps improve luminance (detail) signal, by not contaminating it with cross-channel camera-space RGB and XYZ-space manipulations.

The matrix or channel blend/mapping is selected using the '**Matrix**' parameter. Please note that the available options under this parameter are dependent on the type of dataset you imported. Please use the Compose module to import any narrowband data separately.



Four example renderings of the same S-II + H-alpha + O-III narrowband data. Top: two examples of a 1-click SHO (HST palette) rendering. Bottom: two examples of a OHS rendering. No recompositing is needed, and detail is kept fully intact - only the colouring changed.

## Presets

As in most modules in StarTools, a number of presets are available to quickly dial in useful starting points.

- 'Constancy' sets the default Color Constancy mode and is the recommended mode for diagnostics and color balancing in.
- 'Legacy' switches to a color rendition for visual spectrum datasets that is closest to what legacy software (e.g. software without signal evolution Tracking) would produce. This will mimic the way such software (incorrectly) desturates highlights and causes hue shifts.
- 'SHO(HST)' dials in settings that are a good starting point for datasets that were imported as S-II, H-alpha and O-III for red, green and blue respectively (also known as the 'SHO, 'SOH:RGB', 'HST' or 'Hubble' palette). This standard way of importing datasets and mapping the 3 bands to the 3 channels in this way (via the Compose module), allows for further channel blends and remapping via the '**Matrix**' parameter. Please note the specific blend's parameters/factors under the 'Matrix' parameter. This preset also greatly reduces the green bias to minimise green, while attempting to bring out the popular golden hues.
- 'SHO:OHS' is similar to the 'SHO(HST)' preset, except that it further remaps a SHO-imported dataset to a channel blend that is predominantly mapped as OHS:RGB instead. Renditions typically yield a pleasing ''glowing ice-on-fire'' effect.

- 'Bi-Color' assumes a dataset was imported as HOO:RGB, that is Ha-alpha imported as red, and O-III (sometimes also incorporating H-beta) imported as green and also blue. This yields the popular red/cyan bi-color renderings that are so effective at showing dual emission dominance. This preset is also particularly useful and popular for people who use a duo-band filter (aka as a tri-band or quad-band filter) with an OSC or DSLR.

# Contrast: Local Contrast Optimization

The Contrast module optimises local dynamic range allocation, resulting in better contrast, reducing glare and bringing out faint detail.

It operates on medium to large areas, and is especially effective for enhancing contrast and detail unobtrusively in image-filling nebulas, globular clusters and galaxies.

Top: globally stretched data without further local dynamic range optimisation. Bottom: Large to medium scale local dynamic range optimisation with the Contrast Module.

# Usage

The contrast module works by evaluating minimum and maximum brightness in a pixel's local area, and using these statistics to adjust the pixel's brightness. The size of the local areas is controlled by the '**Locality**' parameter. In essence, the '**Locality**' parameter controls how 'local' the dynamic range optimisation is allowed to be. You will find that a higher '**Locality**' value with all else equal, will yield an image with areas of starker contrast. More generally, you will find that changing the '**Locality**' value will see the Contrast module take rather different decisions on what (and where) to optimise. The rule of thumb is that a higher '**Locality**' value will see smaller and 'busier' areas given priority over larger more 'tranquil' areas.

The '**Shadow Detail Size**' parameter specifies how ''careful'' the Contrast module should be with dark detail. Dark detail below a certain size, may have some of its dynamic range de-allocated and given back a 'reduced' dynamic range allocation. The relative size (in percentage points) of this dynamic range that is given back, is specified by the '**Shadow Dynamic Range Allocation**' parameter. The higher this value, the more dynamic range is optimised for small bright detail and larger dark detail, land less for small dark detail.

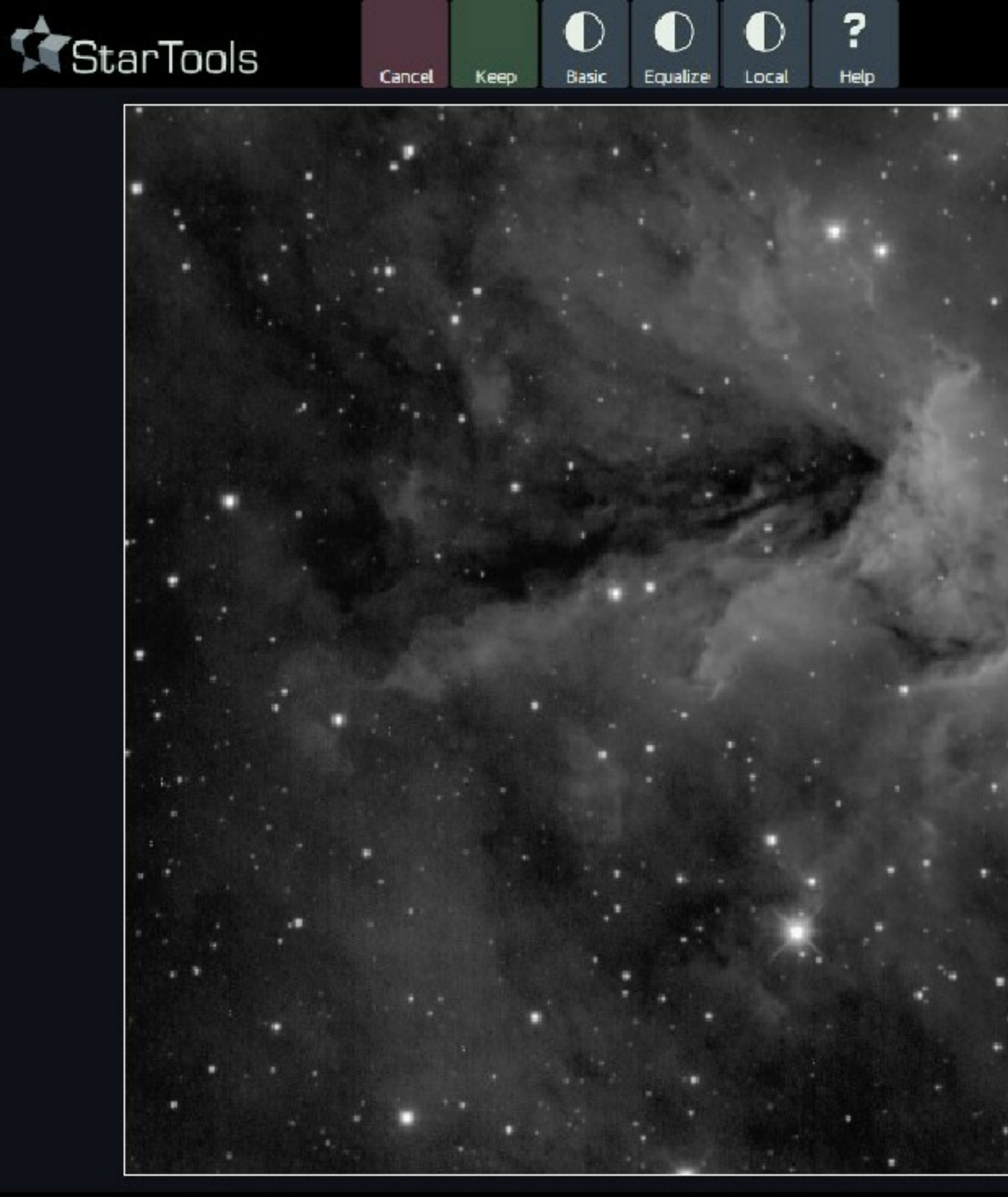
As alluded before, The '**Shadow Dynamic Range Allocation**' parameter controls how heavily the Contrast module ''squashes'' the dynamic range of dark, smaller scale features it deems ''unnecessary''; by de-

allocating dynamic range that is used to describe larger features and re-allocating it to interesting local features, the de-allocation necessarily involves reducing the larger features' dynamic range, hence ''squashing'' that range. Very low settings may appear to clip the image in some extreme cases (though this is not the case). For those familiar with music production, the Contrast module is analogous to a compressor, but for your images instead.

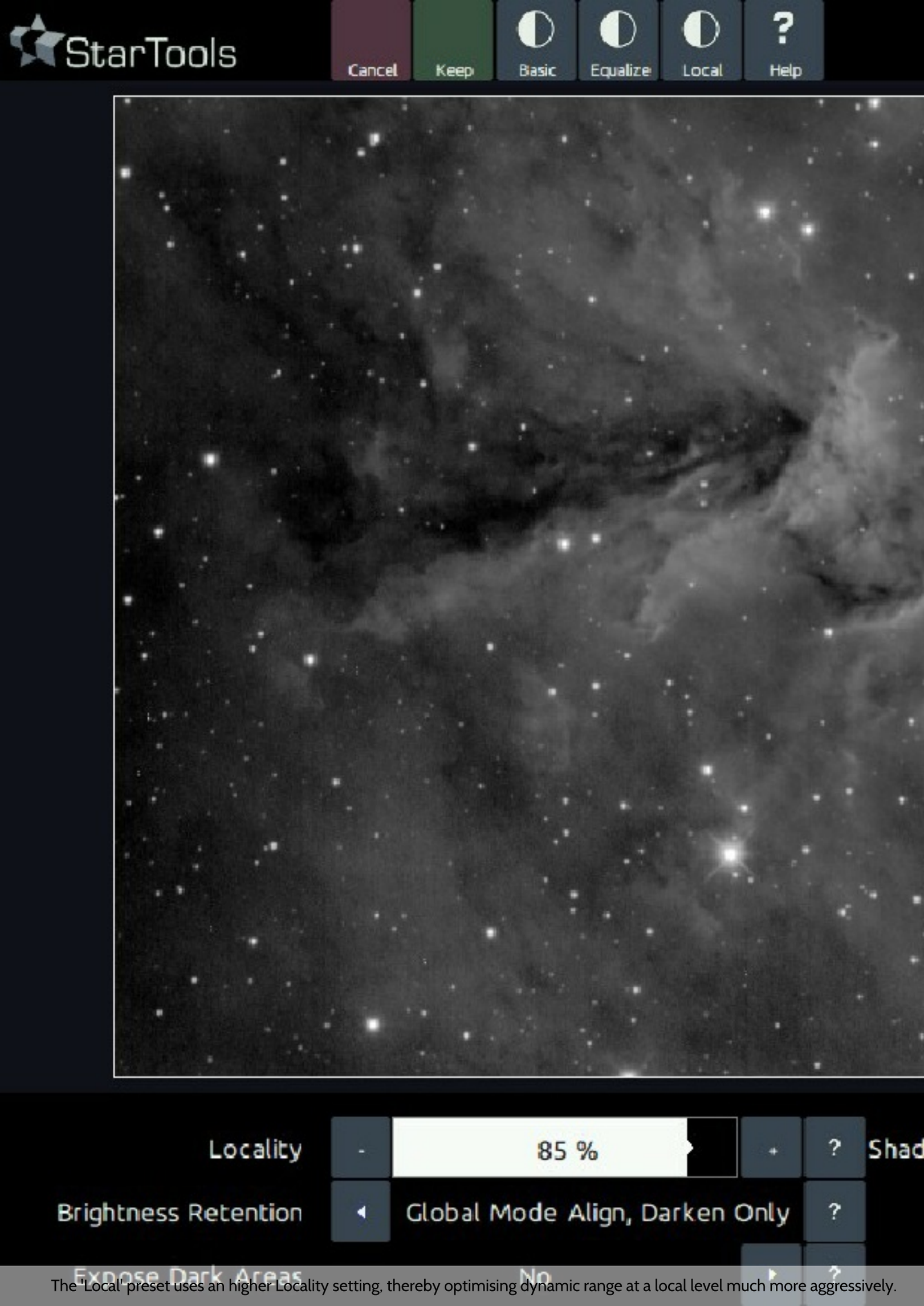
The '**Brightness Retention**' feature attempts to retain the apparent brightness of the input image. It does so through calculating a non-linear stretch that aligns the histogram peak (statistical 'mode') of the old image with that of the new image. An optional 'Darken Only' operation only keeps pixels from the resulting image that are darker than the input image.

We will use this Hydrogen Alpha dataset of Meloitte 15, acquired by Jim Misti to demonstrate the Contrast module. Pre-processing consisted of a simple Wipe and AutoDev.

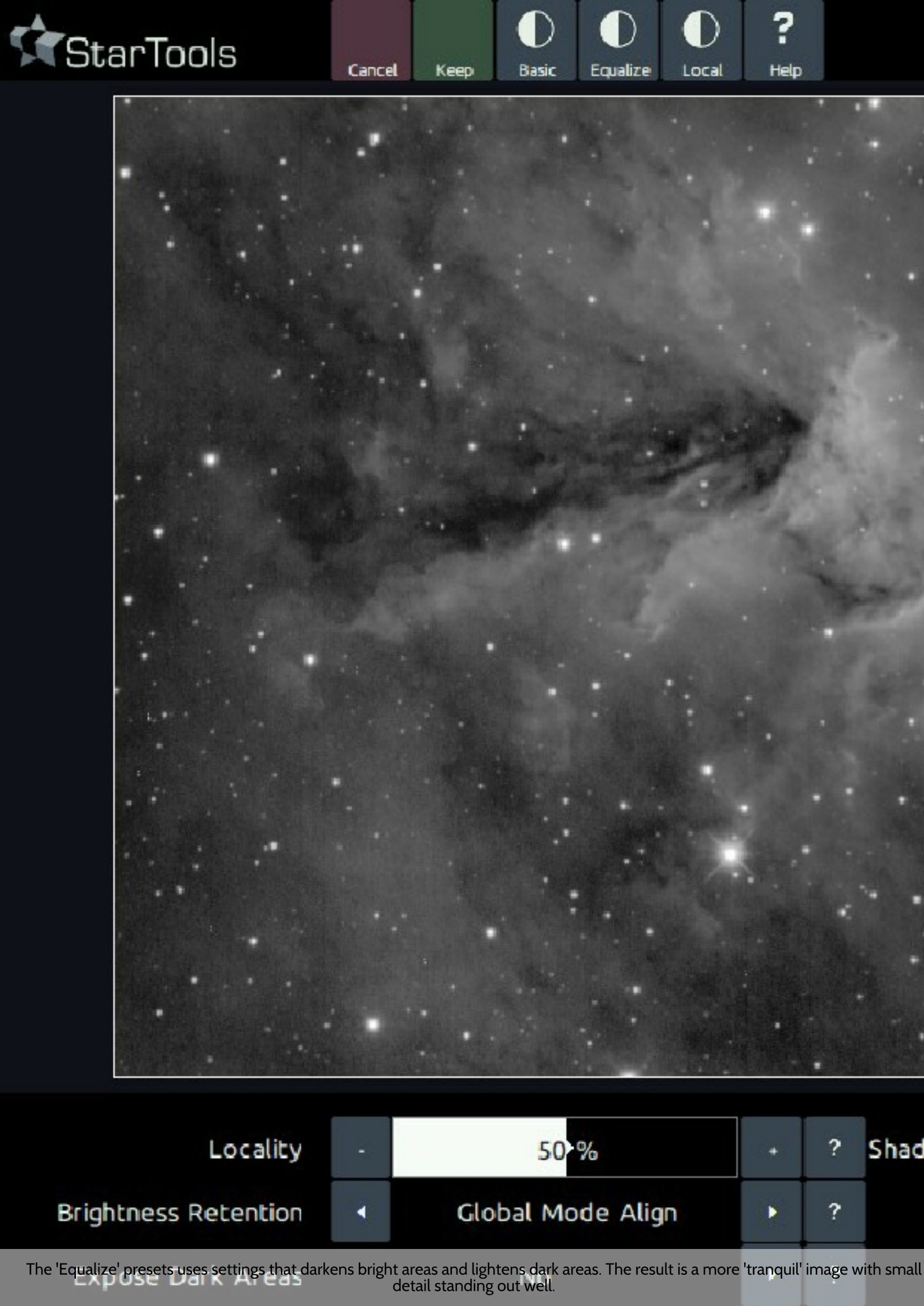
The **'Expose dark areas'** option can help expose detail in the shadows by normalizing the dynamic range locally; making sure that the fully dynamic range is used at all times.




Default settings corresponding with the 'Basic' preset. The image is locally darkened, removing glare.




The 'Local' preset uses an higher Locality setting, thereby optimising dynamic range at a local level much more aggressively.




The 'Equalize' presets uses settings that darkens bright areas and lightens dark areas. The result is a more 'tranquil' image with small detail standing out well.

# Compose: Effortless, Signal Evolution-Tracked Complex Composite Processing

The Compose module is easy-to-use, but extremely flexible compositing and channel extraction tool. As opposed to all other software, the Compose module allows you to effortless process, LRGB, LLRGB, or narrowband composites like SHO, LSHO, Duo/Tri/Quadband, HaLRGB etc. composites, *as if they were simple RGB datasets.*

In traditional image processing software, composites with separate luminance, chrominance and/or narrowband filters, require lengthy processing workflows; luminance (detail), chrominance (color) and narrowband accents datastreams need (or should!) be processed separately and only combined at the end to produce the final image.

Through the Compose module, StarTools is able to process luminance, color and narrowband accent information separately, yet simultaneously.

This has important ramifications for your workflow and signal fidelity;

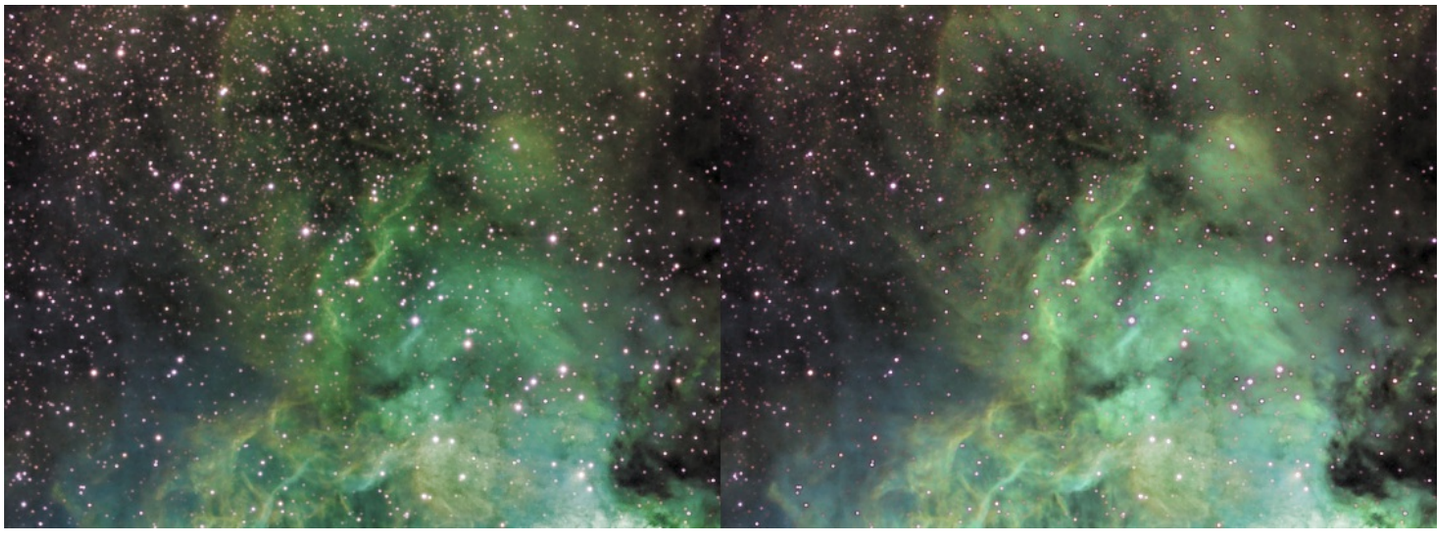
- Your workflow for a complex composite is now

virtually the same as it is for a simple DSLR/OSC dataset; Modules like Wipe and Color automatically consult and manipulate the correct dataset(s) and enable additional functionality where needed.

- Because everything is done in one Tracking session, you get all the benefits from signal evolution tracking until the very end, without having to end your workflow for luminance and start a new one for chrominance or narrowband accents; all modules cross-reference luminance and color information as needed until the very end, yielding vastly cleaner results.
- The "Entropy" module can consult the chroma/color information to effortlessly manipulate luminance as you see fit, while Tracking monitors noise propagation.

Synthetic luminance dataset are created by simply specifying the total exposure times for each imported dataset. With a click of a button, synthetic luminance datasets can be added to an existing luminance dataset, or can be used as a (synthetic) luminance dataset in its own right.

Finally, the Compose module can be used to create bi-color composites, or to extract individual channels from color images.



In conjunction with the Composite module, the Entropy module can be used to boost detail in, for example, Synthetic L+SHO datasets, while availing of signal evolution Tracking. Here O-III was boosted, while Tracking kept noise propagation under control.

# Usage

Creating a composite is as easy as loading the desired datasets into the desired slots, and optionally setting the desired composite scheme and exposure lengths.

Care must be taken that all datasets are of the *exact same* dimensions and are perfectly aligned. Alignment should *always* be done during stacking (by means of a common reference stack) and *never* after the fact when the datasets have already been stacked. Alignment during stacking will yield the least amount of errors in point spread functions and chrominance (color) signal, which is important for operations such as

deconvolution and color calibration.

The "**Luminance**" button loads a dataset into the "**Luminance File**" slot. The "**Lum Total Exposure**" slider determines the total exposure length in hours, minutes and seconds. This value is used to create the correct weighted synthetic luminance dataset, in case the "**Luminance, Color**" composite mode is set to create a synthetic luminance form the loaded channels. Loading a Luminance file will only have an effect when the "**Luminance, Color**" parameter is set to a compositing scheme that incorporates a luminance dataset (e.g. "L, RGB", "L + Synthetic L From RGB, RGB" or "L + Synthetic L From RGB, Mono") .

The "**Red/S-II**", "**Green/Ha**" and "**Blue/O-III**" buttons load a dataset in the "**Red File**", "**Green File**" and "**Blue File**" slots respectively. The "**Red Total Exposure**", "**Green Total Exposure**", "**Blue Total Exposure**" sliders determine the total exposure length in hours, minutes and seconds for each of the three slots. These values are used to create the correct weighted synthetic luminance dataset (at 1/3rd weighting of the "Lum Total Exposure"), in case the "**Luminance, Color**" composite mode is set to create a synthetic luminance from the loaded channels.

The "**NBAccent**" button loads a dataset for parallel processing as narrrowband accents (see NBAccent module).

Loading an dataset into the "**Red File**", "**Green File**" or "**Blue File**" slots will see any missing slots be synthesised automatically if the "**Color Ch. Interpolatio**n" parameter is set to "**On**". Note that loading a colour

dataset into the "**Red File**", "**Green File**" or "**Blue File**" slots will automatically extract the red, green and blue channels of the colour dataset respectively.

Note that the **Red/S-II**, **Green/Ha** and **Blue/O-III** buttons at the top of the module have alternative designations as well, for use when importing "SHO" datasets. In this case, S-II is mapped to the Red channel, H-alpha is mapped to the Green channel, O-III is mapped to the blue channel.

There are a number of compositing schemes available, most of which will put StarTools into "composite" mode (as signified by a lit up "Compose" label on the Compose button on the home screen). Compositing schemes that require separate processing of luminance and colour will put StarTools in this special mode. Some module may exhibit subtly different behaviour, or expose different functionality while in this mode.

The following compositing schemes are selectable;

- "**RGB, RGB (Legacy Software)**" simply uses red + green + blue for luminance and uses red, green and blue for the color information. No special processing or compositing is done. Any loaded Luminance dataset is ignored, as are Total exposure settings. This is how less sophisticated software from years past ("legacy") would composite your datasets.

- "**RGB, Mono**" simply uses red + green + blue for luminance and uses the average of the red, green and blue channels for all channels for the color information, resulting in a mono image. Any loaded Luminance dataset is ignored, as are Total exposure settings.
- "**L, RGB**" simply uses the loaded luminance dataset for luminance and uses red, green and blue for the colour information. Total exposure settings are ignored. StarTools will be put into "composite" mode, processing luminance and colour separately but simultaneously. If not Luminance dataset is loaded, this scheme functions the same as "RGB, RGB" with the exception that StarTools will be put into "composite" mode, processing luminance and colour separately yet simultaneously.
- "**L + Synthetic L from RGB, RGB**" creates a synthetic luminance dataset from Luminance, Red, Green and Blue, weighted according to the exposure times provided by the "Total Exposure" sliders. The colour information will consists of simply the red, green and blue datasets as imported. StarTools will be put into "composite" mode, processing luminance and colour separately yet simultaneously.
- "**L + Synthetic L from RGB, Mono**" creates a synthetic luminance dataset from Luminance, Red, Green and Blue, weighted according to the exposure times provided by the "Total Exposure" sliders. The colour information will consists of the average of the red, green and

blue channels for all channels, yielding a mono image. StarTools is not put into "composite" mode, as no colour information is available.

- "**L + Synthetic L from R(2xG)B, RGB (Color from OSC/DSLR)**" creates a synthetic luminance dataset from Luminance, Red, Green and Blue, weighted according to the exposure times provided by the "Total Exposure" sliders. The green channel's contribution is doubled to reflect the originating instrument's Bayer Matrix having twice the amount of green samples. The colour information will consists of simply the red, green and blue datasets as imported. StarTools will be put into "composite" mode, processing luminance and colour separately yet simultaneously. This mode is suitable for OSC and DSLR datasets and is used internally by the "Open" functionality on the home screen when the user chooses the second option "Linear from OSC/DSLR with Bayer matrix and not white balanced".
- "**L + Synthetic L from RGB, R(GB)(GB) (Bi-Color)**" creates a synthetic luminance dataset from Luminance, Red, Green and Blue, weighted according to the exposure times provided by the "Total Exposure" sliders. The colour information will consists of red as imported, with an average of green+blue assigned to both the green and blue channels. This mode is suitable for creating bi-colours from, for example, two narrowband filtered datasets.
- "**L + Synthetic L from R(2xG)B, R(GB)(GB) (Bi-Color from OSC/DSLR)**" creates a synthetic luminance dataset from Luminance, Red, Green and Blue, weighted according to the exposure times

provided by the "Total Exposure" sliders and taking into account the presence of a Bayer matrix. The colour information will consists of red as imported, with an average of green+blue assigned to both the green and blue channels. This mode is very useful for creating bi-colours from duo/tri/quad band filtered datasets.

## On synthetic luminance generation

For practical purpose, synthetic luminance generation assumes that, besides possibly varying total exposure lengths, all other factors remain equal. E.g. it is assumed that bandwidth response is exactly equal to that of the other filters in terms of width and transmission, and that *only* shot noise from the object varies (either due to differences in signal in the different filter band from the imaged object, or due to differing exposure times).

When added to a real (non synthetic) luminance source (e.g. the optional source imported as 'Luminance File'), the synthetic luminance's three red, green and blue channels are assumed to contribute exactly one third to the added synthetic luminance. E.g. it is assumed that the aggregate filter response of the individual three red, green and blue channels, exactly match that of the single 'Luminance File' source. In other words, it is assumed that;

```
red filter response + green filter response + blue
filter response = luminance filter
```

`response`

If the above is not (quite) the case and your know exact filter permeability, you can prorate the filter response by varying the Total Exposure sliders.

Finally, in the case of the presence of an instrument with a Bayer matrix, the green channel is assumed to contribute precisely 2x more signal than the red and blue channels.

Any narrowband accent data loaded, does not impact synthetic luminance generation.

## Channel assignment and coloring and narrowband datasets

Unique to StarTools, **channel assignment does not dictate final coloring**. In other words, loading, for example, a SHO dataset as RGB, does not lock you into using precisely that channel mapping. Thanks to the signal evolution Tracking engine, the Color module allows you to completely remap the channels at will for the purpose of colouring, even far into your processing.

As is common practice in astronomy, StarTools assumes channels are imported in order of descending wavelength. E.g. the dataset with the longest wavelength (e.g. the light with the highest nm or Å comes first). In other words, the reddest light comes first, and the bluest light comes last.

In practice this means that;

- When using visual spectrum datasets, load red into the red channel, green into the green channel, and blue into the blue channel.
- When using triple channel narrowband datasets such as Hubble-like S-II + H-alpha + O-III (aka "SHO" datasets), load S-II as red, H-alpha as green and O-III as blue.
- When using a duo/tri/quad band filtered dataset, load H-alpha (which is possibly combined with the neighbouring S-II line depending on the filter) as red, and load O-III (which is possibly combined with the neighbouring H-beta line depending on the filter) as green.

In any case, you should not concern yourself with the colouring until you hit the Color module in your workflow; as opposed to other software, this initial channel assignment has no bearing *at all* on the final colouring in your image. Please note that failing to import channels correctly in the manner and order described above, will cause the Color module to mis-label the many colouring and blend options it offers.

## Narrowband accents

With the introduction of the NBAccent module in StarTools 1.8, a third parallel datastream type has been introduced; that of narrowband accents for visual spectrum augmentation. Adding narrowband accents to visual spectrum datasets has traditionally been a daunting, difficult and laborious process, involving multiple workflows. The NBAccent module is a powerful module that starts its work as soon as your load your data in the Compose module. Crucially it adds only a *single*, easy step to an otherwise standard workflow, while yielding superior results in terms of color fidelity/preservation.

By making narrowband accents an integral part of the complete workflow and signal path, results are replicable, predictable and fully tracked by StarTools' unique signal evolution Tracking engine, yielding perfect noise reduction every time.

Enabling narrowband accents in your wofklow, is as easy as loading the file containing the signal you wish to add as narrowband accents, and specifying the type of accents the file contains. Three possible types are selectable;

- H-alpha or S-II from a narrowband filter
- O-III or H-beta from a narrowband filter
- A combination of narrowband signals across

multiple channels from a duo, tri or quadband filter (such as the Optolong L-Extreme or L-eNhance) or a combined single narrowband filter

Dedicated functionality for visual spectrum narrowband accents are part of StarTools' integrated workflow.

Be sure to specify the correct type before continuing.

## Popular coloring

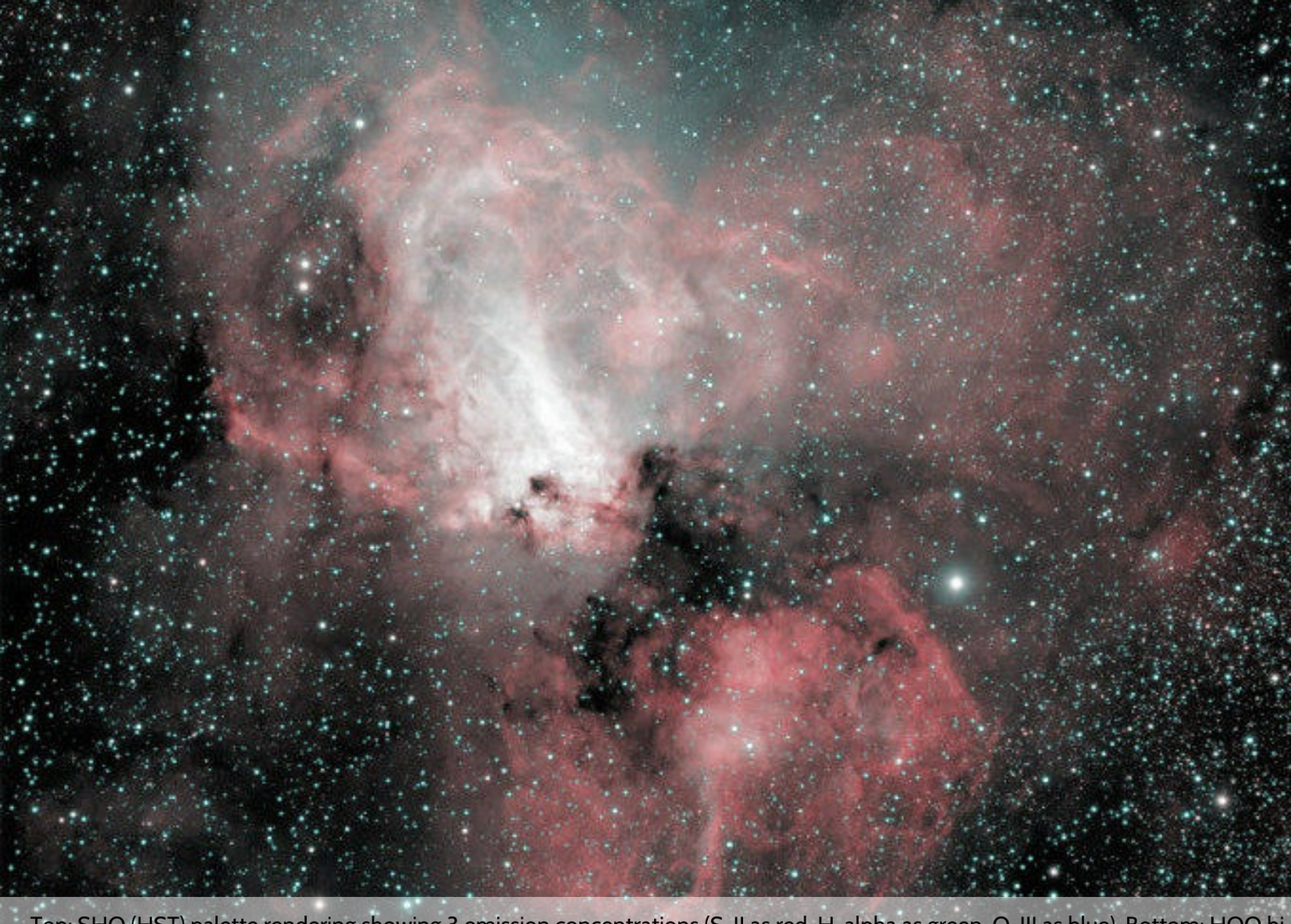
### Popular narrowband composite colouring

#### Hubble / HST / SHO

The Hubble Space Telescope palette (also known as 'HST' or 'SHO' palette) is a popular palette for color renditions of the S-II, Hydrogen-alpha and O-III emission bands. This palette is achieved by loading S-II, Hydrogen-alpha and O-III (''SHO'') as red, green and blue respectively. A special ''**Hubble**'' preset in the Color module provides a shortcut to color rendition settings that mimic the results from the more limited image processing tools from the 1990s.

#### H-alpha + O-III bi-color

A popular bi-color rendition of H-alpha and O-III is to import H-alpha as red and O-III as green as well as blue. A synthetic luminance frame is then created that only gives red and blue (or green *instead of* blue, but not both!) a weighting according to the two datasets' exposure lengths. The resulting color rendition tends to be close to these bands' manifestation in the visual spectrum with H-alpha a deep red and O-III appearing as a teal green.

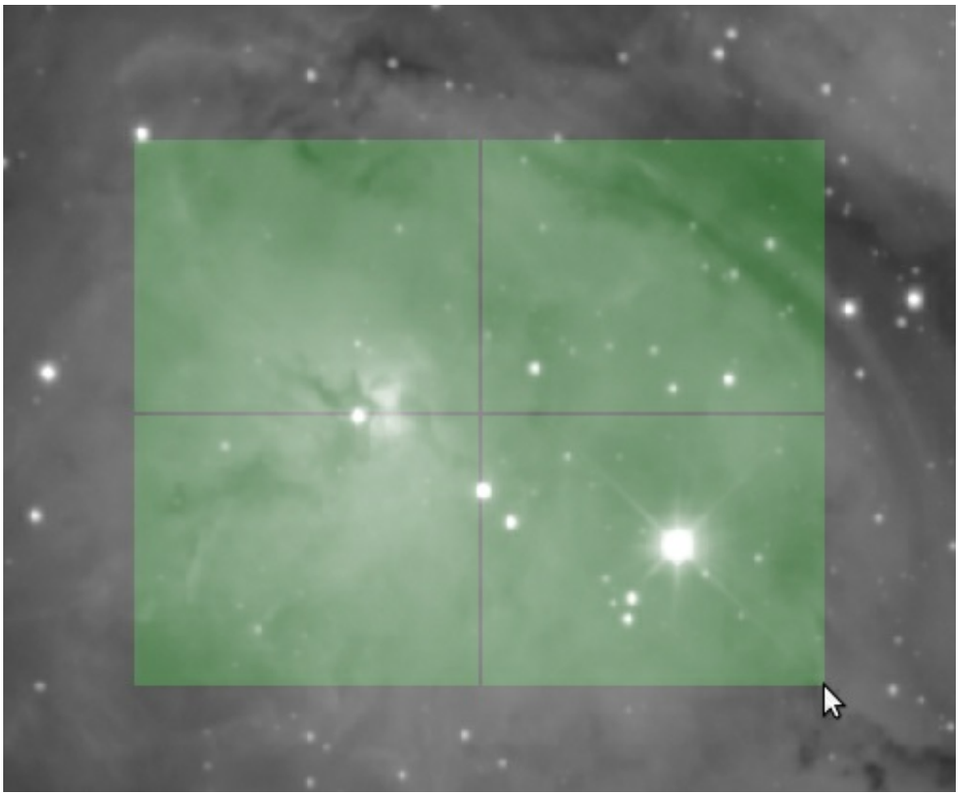


Top: SHO (HST) palette rendering showing 3 emission concentrations (S-II as red, H-alpha as green, O-III as blue). Bottom: HOO bi-color rendering showing 2 emission concentrations (Ha-alpha as red, O-III as cyan).

# Crop: Express Cropping Tool with Switchable Luminance, Chrominance and Narrowband Accent Previewing

The crop module is an easy-to-use image cropping tool with quick aspect ratio presets and switchable luminance, chrominance and narrowband accent preview modes.

The module was designed to quickly find and eliminate stacking artefacts across luminance, chrominance and narrowband accent data, as well as help with framing your object(s) of interest.



Cropping is as easy as clicking and dragging with the mouse

# Usage

Using the crop module is fairly straightforward. The desired crop is created by clicking and dragging with the mouse the area to retain. Fine-tuning can be accomplished by changing the X1, Y1 and X2, Y2 coordinate pair parameters.

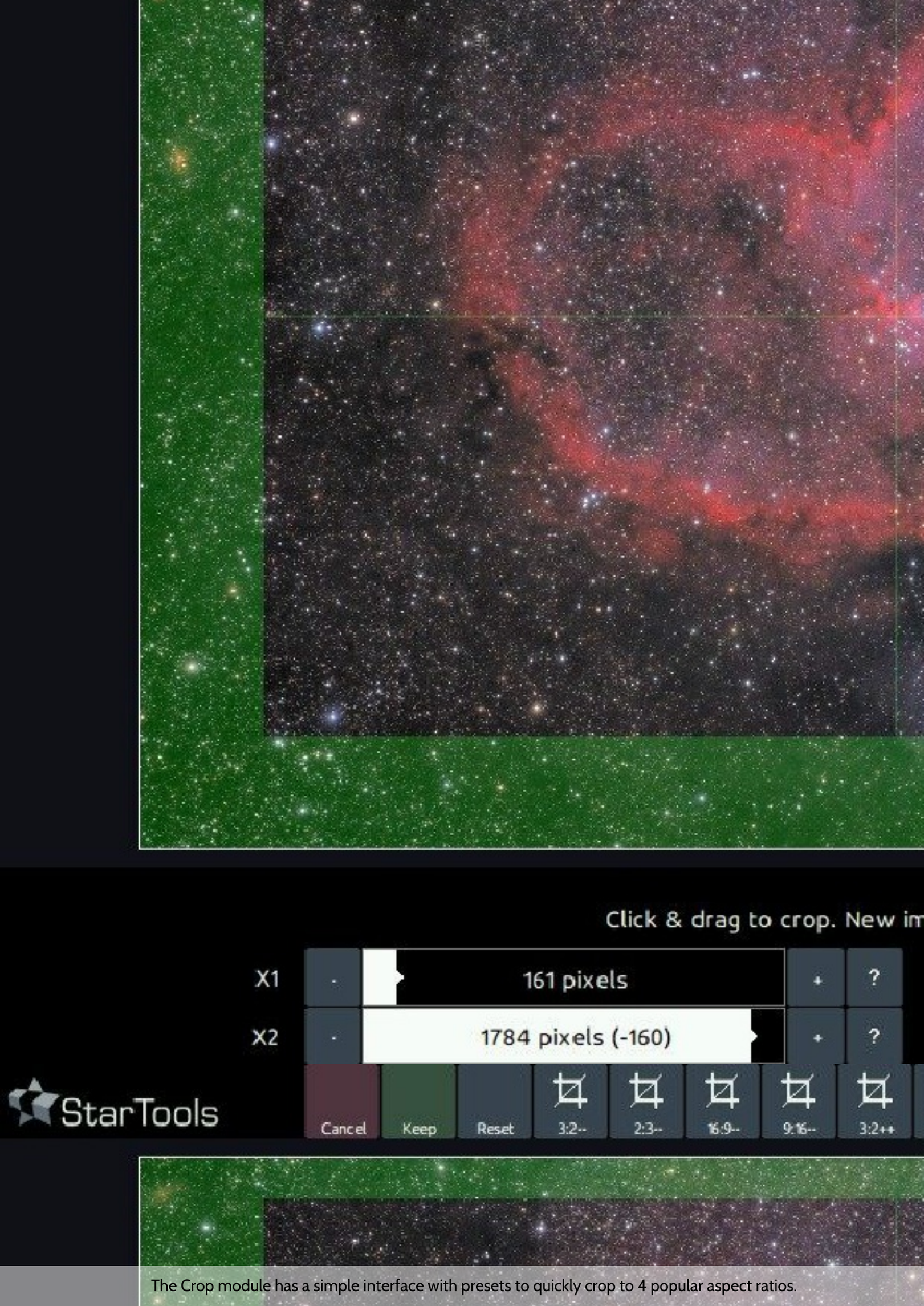
8 quick-access crops are available to quickly achieve one of four popular aspect ratios. The button names ('3:2', '2:3', '16:9', 9:16') denote the aspect ratio, while the double minus ('--') or plus ('++') signs postfix denotes their behaviour;

- Buttons with the '--' postfix will shrink the current

selection to achieve the selected aspect ratio

- Buttons with the '++' postfix will grow the current selection to achieve the selected aspect ratio

A '**Color**'/'**NBAccent**' button is available, which functions much like the '**Color**'/'**NBAccent**' button in the Wipe module. Like in the Wipe module, it is only available when Compose mode is engaged (e.g. when luminance, chrominance and/or narrowband accents are being processed separately, yet simultaneously). The button allows you to switch the view between the luminance, chrominance and narrowband accent datasets that are being processed in parallel. The later is useful if, for example, you need to crop stacking artefacts that only exist in the chroma dataset and/or narrowband accent dataset, but not in the luminance dataset. Because chrominance data always remains linear and is never stretched like the luminance dataset, a courtesy (non-permanent) AutoDev is applied, so you can better see what is in the chrominance dataset. Likewise, a courtesy temporary AutoDev is applied to any narrowband accent data for that same purpose.




The Crop module has a simple interface with presets to quickly crop to 4 popular aspect ratios.

# Unified De-Noise: Detail Aware Wavelet-based Noise Reduction and Noise Grain Shaper

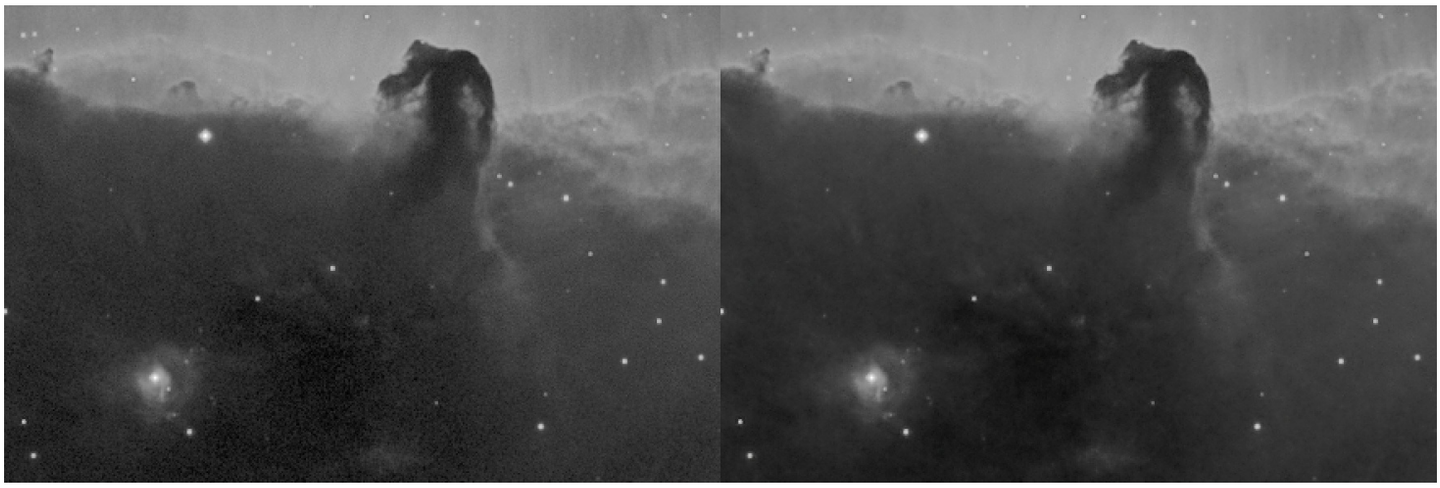
The Unified De-Noise module offers *temporal*, astro-specific noise reduction. Paired with StarTools' Tracking feature, it yields pin-point accurate results that have no equal.

The Unified De-Noise module is the ultimate application of the signal evolution Tracking feature (which data mines every decision and noise evolution per-pixel during the user's processing). The results that Unified De-Noise is able to deliver autonomously because of this deep knowledge of your signal and its evolution, are absolutely unparalleled; like many algorithms in StarTools, the algorithm works on a temporal (3D) basis, rather than just spatial, giving it vastly more data to work with.

Whereas generic noise reduction routines and plug-ins for terrestrial photography are often optimised to detect and enhance geometric patterns and structures in the face of random noise, the Unified De-Noise module is created to do the opposite. That is, it is careful *not* to enhance structures or patterns, and instead attenuates the noise and gives the user control over its appearance. Its unified noise reduction routines are specifically designed to be "permissible" even for scientific purposes - that is, it was designed to only carefully remove energy from

the image and not add it; it strictly does not sharpen, edge-enhance or add new "detail" to the image.

In addition, StarTools is currently the only software that can be made to also specifically target walking noise (streaks) caused by not being able to dither during acquisition (for example when conducting Electronically-Assisted Astronomy).



200% zoom showing pin-point accurate, autonomous, fully configurable noise reduction based on data mined statistics. No masks, no other subjective crutches.

# Usage

Denoising starts when switching Tracking off. It is therefore the last step in your workflow, and for good reason; being the last step, Tracking has had the longest possible time to track and analyse noise propagation.

## Setup

The first stage of noise reduction involves helping StarTools establish a baseline for visual noise grain and the presence (and direction) of walking noise. To establish this baseline, increase the '**Grain size**' parameter until no noise grain of any size can be seen any longer. StarTools will use this baseline as a guide as to what range of details in your image is affected by visible noise.

If walking noise is present, then temporarily set '**Grain Size**' parameter to 1.0. Next, use the '**Walking Noise Angle**' level setter, or click & drag an imaginary line on the image in the direction of the walking noise to set the '**Walking Noise Angle**' that way. Now increase the '**Walking Noise Size**' parameter until individual streaks are no longer visible in the direction you detected them in (though other imperfections may still be visible). After that, increase the 'Grain Size' parameter until other noise grain can no longer be seen.

After clicking 'Next', analysis and wavelet scale extraction starts, upon which, after a short while, the second interactive noise reduction stage interface is presented.

StarTools is the only software solution for astrophotography that can target walking noise.

# Main operation

Noise reduction and grain shaping is performed in three stages.

## Stage one

The first-pass algorithm is an enhanced wavelet denoiser, meaning that it is able to attenuate features based on their size. Noise grain caused by shot noise (aka Poisson noise) - the bulk of the noise astrophotographers deal with - exists on all size levels, becoming less noticeable as the size increases. Therefore, much like the Sharp module, a number of scale sizes ('**Scale *n***' parameters) are available to tweak, allowing the denoiser to be more or less aggressive when removing features deemed noise grain at different sizes. Tweaks to these scale parameters are generally not necessary, but may be desirable if - for whatever reason - noise is not uniform and is more prevalent in a particular scale.

Different to basic wavelet denoising implementations, the algorithm is driven by the per-pixel signal (and its noise component) evolution statistics collected during the preceding image processing. E.g. rather than using a single global setting for all pixels in the image, StarTools' implementation uses a different setting (yet centred around a user-specified global setting) *for every pixel* in the image.

The wavelet denoising algorithm is further enhanced by a '**Scale Correlation**' feature parameter, which exploits common psychovisual techniques, whereby noise grain is generally tolerated better in areas of

increased (correlated) detail.

The general strength of the noise reduction by the wavelet denoiser, is governed by the '**Brightness Detail Loss**' and '**Color Detail Loss**' for luminance (detail) and chrominance (colour) respectively.

The noise reduction solution in StarTools is based wholly around energy removal - that is attenuation of the signal and its noise components in different bands in the frequency domain - and avoids any operations that may *add* energy. It does not enhance edges, does not manipulate gradients, and does not attempt to reconstruct detail. These important attributes make its use generally permissible for academic and scientific purposes; it should never suggest details or features that were never recorded in the first place.

## Stage two

Any removed energy, is collected per pixel and re-distributed across the image in a second pass, giving the user intuitive control, via the '**Grain Dispersion**' parameter, over a hard upper size limit beyond which grain is no longer smoothed out.

## Stage three

The '**Grain Equalization**' parameter lets the user *reintroduce* removed noise grain in a modified, *uniform* way, that is; appearing of equal magnitude across the image (rather than being highly dependent per-pixel signal strength, stretches and local enhancements as seen in the input image).

The '**Grain Equalization**' feature an acknowledgement of the ''two schools'' of noise reduction prevalent in astrophotography; there are those who like smooth images with little to no noise grain visible, and there are those who find a tightly controlled, uniform measure of noise grain desirable for the purpose of creating visual interest and general aesthetics (much like noise grain is added for a ''filmic'' look in CGI). The noise signature of the deliberately left-in noise, is precisely shaped to be aesthetically pleasing for precisely this purpose.

Lastly, it should be noted that the '**Grain Equalization**' feature only shapes and re-introduces noise in the luminance portion of the signal, but not in the chrominance (color) portion of the signal.

Left: input image. Middle: Denoised image without Grain Equalization. Right: Denoised image with Grain Equalization. Notice the vast differences in noise grain prevalence independent of brightness, and notice subtle pre-existing noise grain has been re-introduced in an unobtrusive and visually pleasing manner.

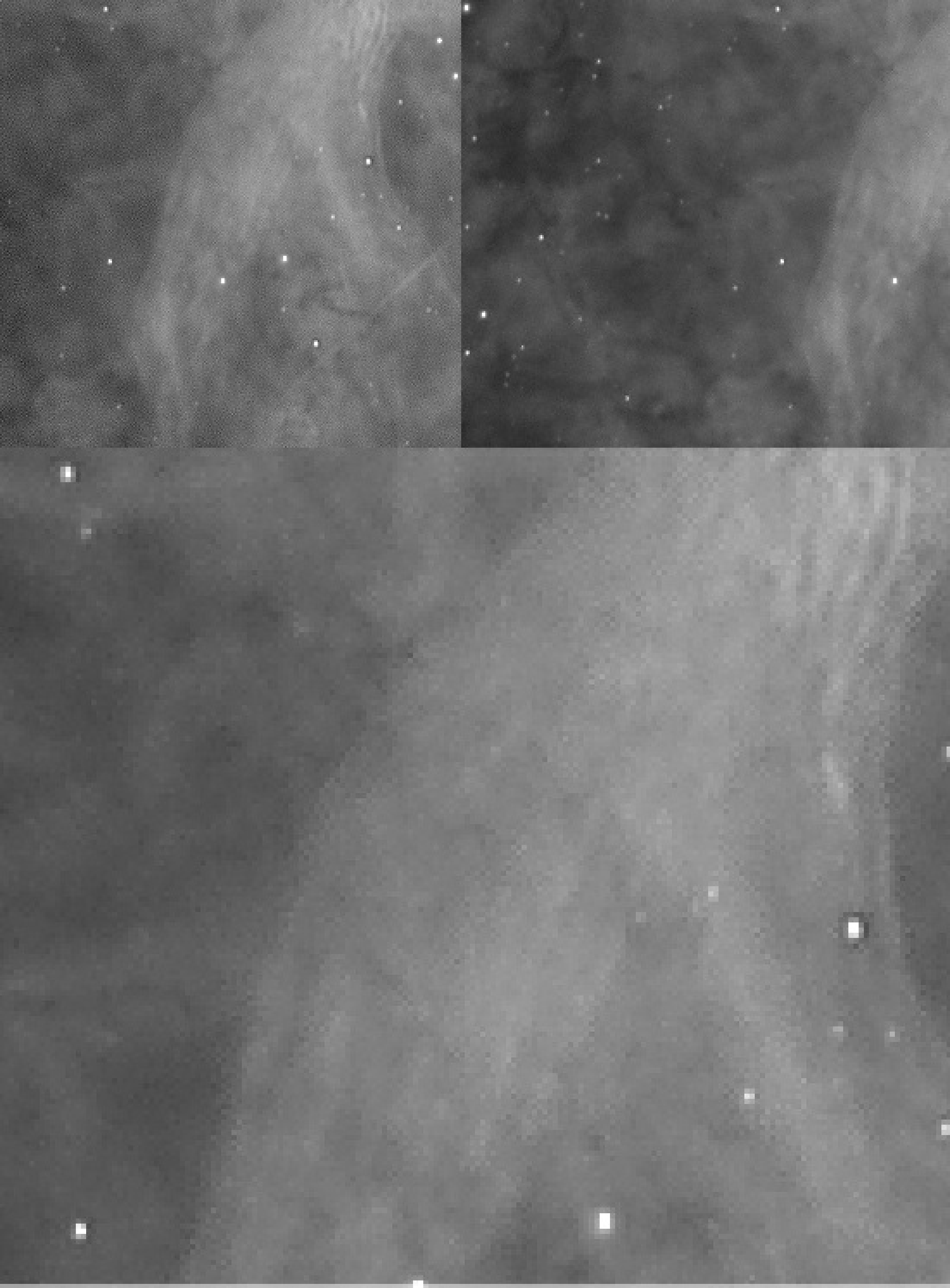
## Evaluating the result

Given StarTools' general design goal of exploiting psychovisual limitations of the human visual system, there are some important things to take note of when evaluating the result.

Specifically, the module exploits "useful" noise grain (by modelling it as quantization error in the signal) to retain and convey more detail in areas that are too "busy" for the human visual system to notice, without the result *appearing* noisier. The actual "useful" noise grain, much like dithering, however may be visible when zoomed in at scales beyond 100%.

The value of the module's ability to shape noise grain in this way, becomes particularly apparent when combining this ability with the output of StarTools' deconvolution module. The latter module can be "overdriven" to trade increased detail for increased (though perceptually equalised) fine grain noise "artifacts". The magnitude of the noise grain is subsequently recovered, modeled and shaped for use as quantization error diffusion in the final denoised image.

Of course, if so desired, using more aggressive parameter settings will progressively eliminate such quantization error diffusion, and yield a smooth image.



Top left: noisy image at 100% zoom. Top right: denoised image at 100%, with grain shaped into quantization error diffusion to retain more detail, even though the image appears to be denoised. Bottom: a 300% zoom of the top right image, revealing the shaped grain

# Inter-channel Entropy-driven Detail Enhancement

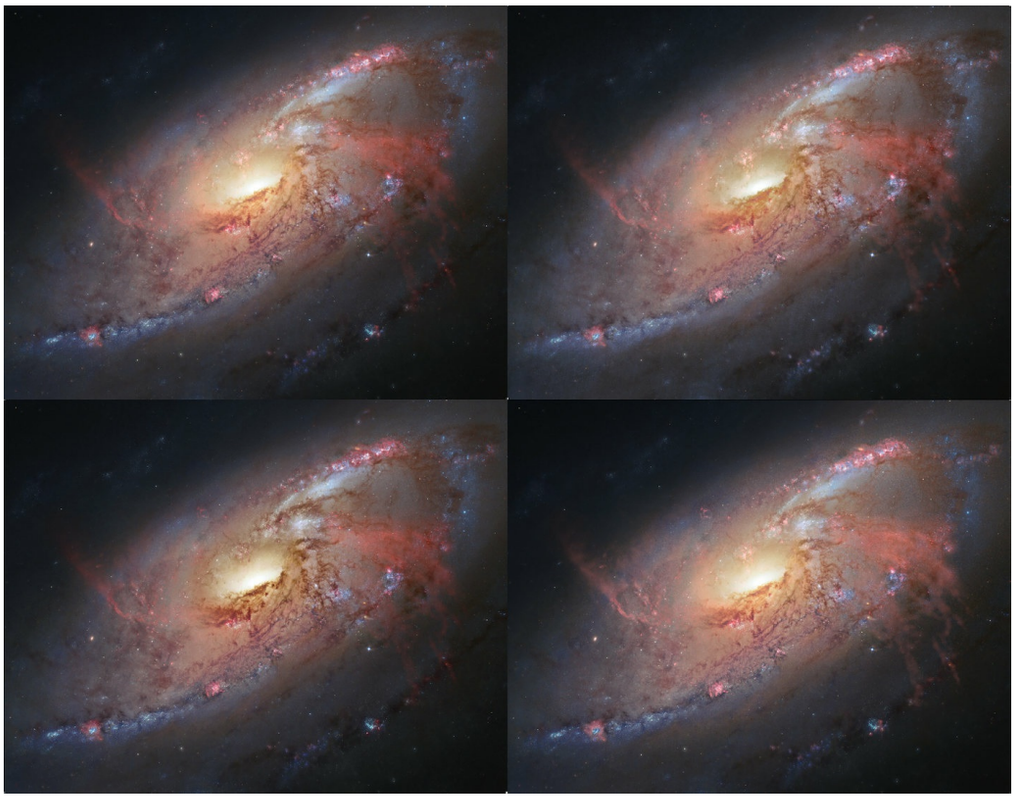
The Entropy module is a novel module that enhances detail in your image, using latent detail cues in the *color* information of your dataset.

The Entropy module exploits the same basic premise as the Filter module; that is, the observation that many interesting features and objects in outer space have distinct colors, owing to their chemical make up and associated emission lines. This correlation become 100% when considering a narrowband composite, where each channel truly is made up of data from distinct parts of the spectrum.

The Entropy module works by evaluating entropy (a measure of "busyness" or "randomness") as a proxy for detail. It does so on a local level in each colour channel for each pixel. Once this measure has been established for each pixel, the individual channel's contribution to luminance for each pixel is re-weighted in CIELab space to better reflect the contribution of visible detail in that channel.

The result is that the luminance contribution of a channel with less detail in a particular area is attenuated. Conversely, the luminance contribution of a channel with more detail in a particular area is boosted. Overall, this has the effect of accentuating latent structures and detail in a very natural manner. Operating entirely in CIELab space means that,

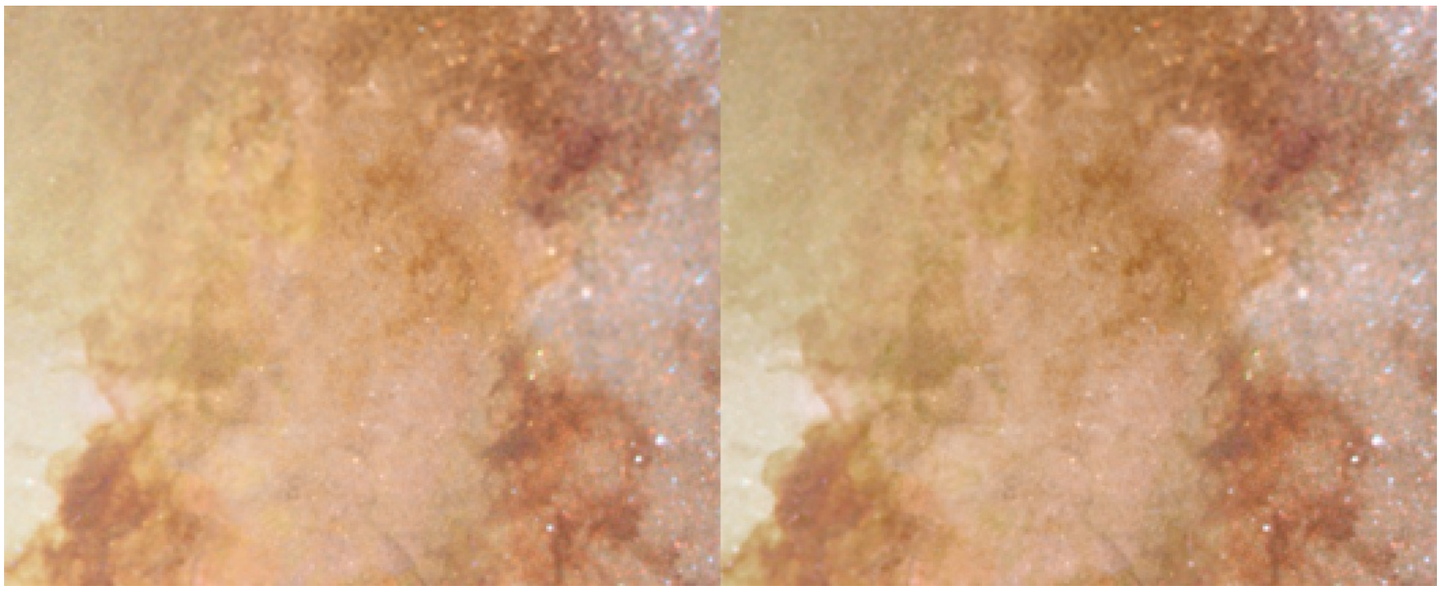
psychovisually, there is no change in colour, only brightness.



Top left: original, top right: all channel optimisation, bottom left: blue channel optimisation, bottom right: red + green optimisation. Original image courtesy of NASA, ESA, the Hubble Heritage Team (STScI/AURA), and R. Gendler (for the Hubble Heritage Team). Acknowledgment: J. GaBany

The above attributes make the Entropy module an an extremely powerful tool for narrowband composites in particular.

The Entropy module is effective both on already processed images, as well as Tracked datasets.



200% zoom detail from NASA 106 image. Left: original, right: Entropy module processed image. Very subtle difference in clarity and contrast can be spotted.

Subtle large-scale structure enhancement. Left: original, middle: Entropy module processed, right: difference map. Of course, the strength of the effect is wholly decided by the user.

# Usage

The Entropy module is very flexible in its image presentation. To start using the Entropy module, an entropy map needs to be generated by clicking the '**Do**' button. This map's resolution/accuracy can be chosen by using the '**Resolution**' parameter. The 'Medium' resolution is sufficient in most cases.

For the entropy module to be able to identify detail, the dataset should ideally be of an image-filling object or scene.

After obtaining a suitable entropy map, the other parameters can be tweaked in real-time;

The '**Strength**' parameter governs the overall strength of the boost or attenuation of luminance. Overdriving the '**Strength**' parameter too much may make channel transitions too visible. In this case you may wish to pull back, or increase the '**Midtone Pull Filter**' size to achieve a smoother blend.

The '**Dark/Light Enhance**' parameter enables you to choose the balance between darkening and brightening of areas in the image. To only brighten the image (for example if you wish to bring out faint H-alpha, but nothing else), set this parameter to 0%/ 100%. To only darken the image (for example to better show a bright DSO core) bring the balance closer to 100%/0%.

The '**Channel Selection**' parameter allows you to only target certain channels. For example, if you wish to enhance S-II more visible in a Hubble-palette image, set this parameter to red (to which S-II should be mapped). S-II will now be boosted, and H-alpha and O-III will be pushed back where needed to aid S-II's contrast. If you wish to avoid the other channels being pushed back, simply set the '**Dark/Light Enhance**' to 0/100%.

The '**Midtone Pull Filter**' and '**Midtone Pull Strength**' parameters, assist in keeping any changes in the brightness of your image confined to the area where they are most effective and visible; the midtones. This feature can be turned off by setting '**Midtone Pull Strength**' to 0%. When on, the filter selectively accepts or rejects changes to pixels, based on whether they are close to half unity (e.g. neutral gray) or not. This feature works analogous to creating a HDR composite from different exposure times. The transition boundaries between accepted and rejected pixels are smoothened out by increasing the '**Midtone Pull Filter**' parameter.

The Entropy module makes for a fantastic narrowband manipulaton tool; here it was used to effortlessly boost the prevalence of O-I emissions in a SHO-mapped image.

# FilmDev: Stretching with Photographic Film Development Emulation

The FilmDev module was created from the ground up as a robust equivalent to the classic Digital Development algorithm that attempts to emulate classic film response when first developing a raw stacked image.

The FilmDev module effectively functions as a classic digital dark room where your prized raw signal is developed and readied for further processing.

The module can also be used as swiss pocket knife for gamma correction, normalisation and channel luminance contribution remixing.

FilmDev stretches your linear image with a bold non-linear stretch that emulates the response of old school photographic film, including all its perks and shortcomings.

# Usage

First off, please note that this module emulates many aspects of photographic film, including its shortcomings. These shortcomings include photographic film's tendency to ''bloat'' stellar profiles. If your goal is to achieve a non-linear stretch that shows as much detail as possible, the far more advanced AutoDev will always do an objectively better job for that purpose. Please note that the edge-enhancing qualities of photographic film are not emulated by this module, as this step is best done through other means.

Enhancements over the classic Digital Development algorithm (Okano, 1997), are the introduction of an additional gamma correction component, the removal of the edge enhancement component, and the introduction of automated black and white point detection. The latter ensures your signal never clips, while making histogram checking a thing of the past.

Central to the module, is the '**Digital Development**' parameter, which controls the strength of the development and resulting stretch. A semi-automated 'homing in feature' attempts to find the optimal settings that bring out as much detail as possible, while still adhering to the Digital Development curve. This feature can be accessed by clicking on the 'Home In' button until the image does not change much further. A simple '**Gamma**' correction can also be applied.

A '**Dark Anomaly Filter**' helps the automatic black point detector ignore any dead pixels. Any dead or darker-than-real-background pixels are caught by the filter, they are re-allocated a reduced amount of dynamic range as set by the '**Dark Anomaly Headroom**' parameter.

Automatic white point detection ('**White Calibration**') uses any over-exposing stars oir other highlights in your image, however it can also be switched to use the '**Dark Anomaly Filter**' setting to filter out any bright anomalies (e.g. hot pixels) that are not stars or real highlights.

An artificial pedestal value can be introduced through the '**Skyglow**' parameter. This parameter specifies how much of the dynamic range (up to 50%) should be taken up by the artificial pedestal.

Finally, a luminance mixer allows for re-mixing of the contribution of each color channel to brightness.

| | | | | |
|---|---|---|---|---|
| Digital Development | - | 91.80 % | + | Home In | ? |
| Gamma | - | 1.00 | + | ? |
| Dark Anomaly Filter | - | 5.0 pixels | + | ? |
| Dark Anomaly Headroom | - | 5 % | + | ? |
| White Calibration | | Use Stars | ▸ | ? |

StarTools

Cancel | Keep | Reset | ? Help

Photogra

## Color retention

Non-linearly stretching an image's RGB components causes its hue and saturation to be similarly stretched and squashed. This is often observable as "washing out" of colouring in the highlights.

Traditionally, image processing software for astrohptography has struggled with this, resorting to kludges like "special" stretching functions (e.g. ArcSinH) or Color enhancement extensions to the DDP algorithm (Okano, 1997) that only attempt to minimize the problem, while still introducing color shifts

While other software continues to struggle with color retention, StarTools Tracking feature allows the Color module to go back in time and completely reconstruct the RGB ratios as recorded, *regardless* of how the image was stretched.

This is one of the major reasons why running the Color module is preferably run as one of the last steps in your processing flow; it is able to completely negate the effect of any stretching - whether global or local - may have had on the hue and saturation of the image.

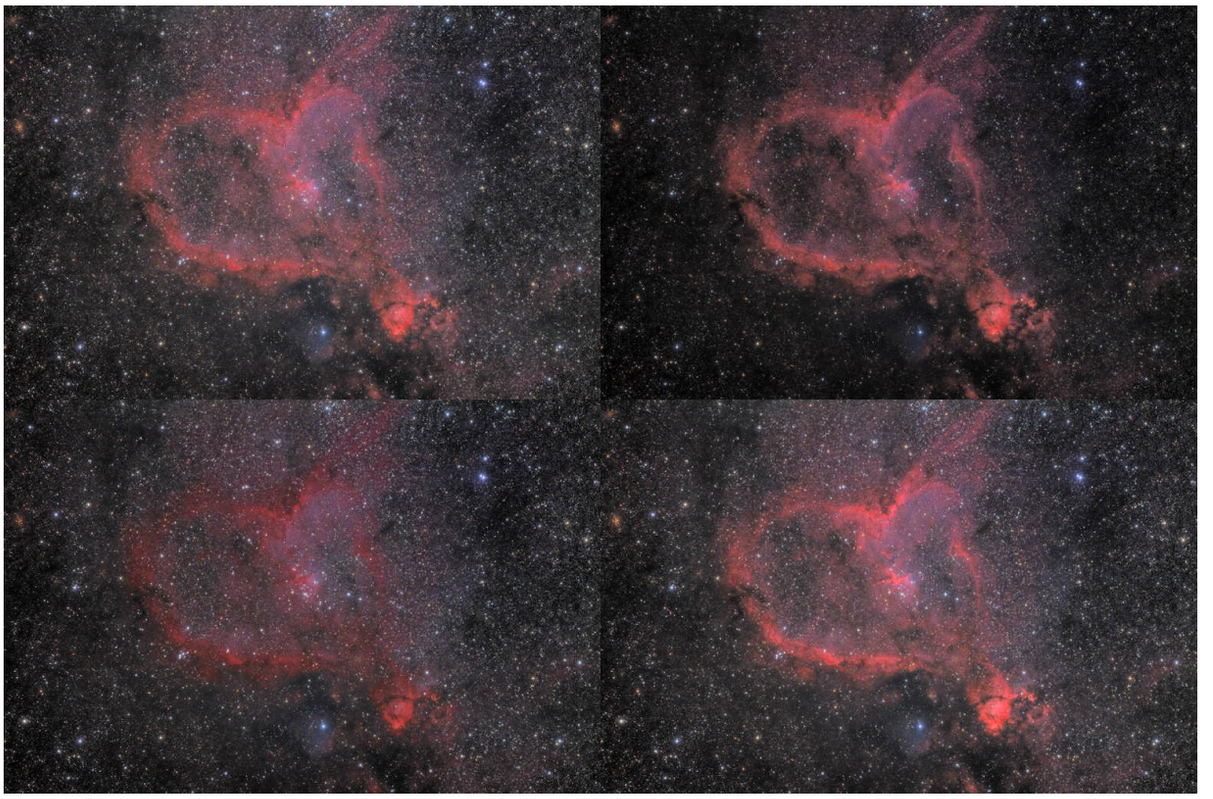
Because of this, the digital development color treatment extensions as proposed by Okano (1997) has not been incorporated in the FilmDev module. The two aspects - colour and luminance - of your image are neatly separated thanks to StarTools' signal evolution Tracking engine.

# Filter: Feature Manipulation by Colour

The Filter module allows for the modification of features in the image by their colour by simply clicking on them. It is as close to a post-capture colour filter wheel as you can get.

Filter can be used to bring out detail of a specific colour (such as faint Ha, Hb, OIII or S2 details), remove artefacts (such as halos, chromatic aberration) or isolate specific features. It functions as an interactive colour filter.

The Filter module is the result of the observation that many interesting features and objects in outer space have distinct colours, owing to their chemical make up and associated emission lines. Thanks to the Color Constancy feature in the Color module, colours still tend to correlate well to the original emission lines and features, despite any wideband RGB filtering and compositing. The Filter module was written to capitalise on this observation and allow for intuitive detail enhancement by simply clicking different parts of the image with a specific colour.



A simple demonstration of the Filter module. All images generated by selecting a filter mode and clicking twice on a part of the red H-alpha region. Top Left: Source. Top Right: Pass. Bottom Left: Reject. Bottom Right: Nudge.

# Usage

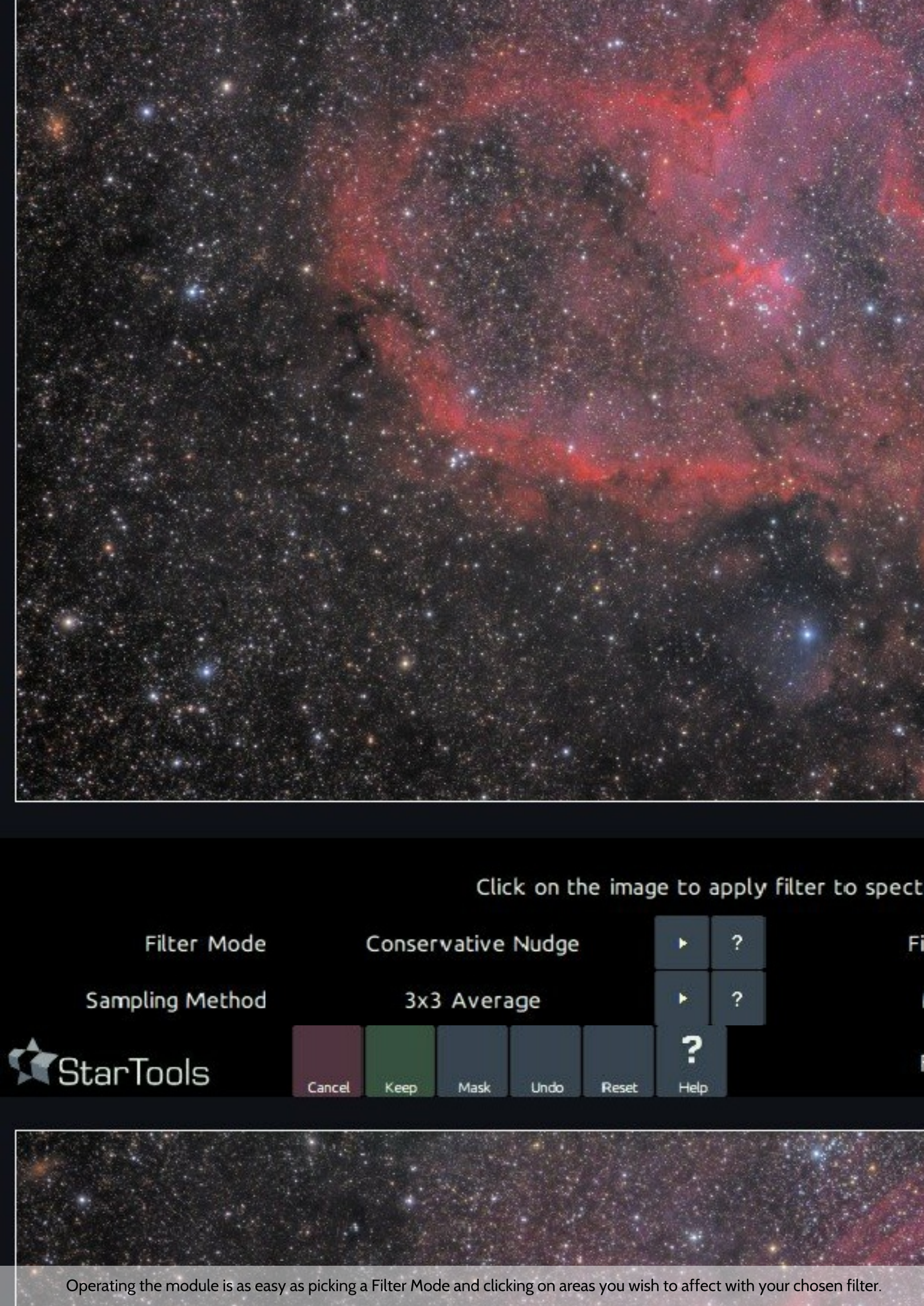
A '**Filter Mode**' parameter selects the mode of the filter. Available modes are;

- 'Conservative Nudge'; this mode boosts the selected signal linearly, but only if the boost would not yield any overexposure
- 'Nudge (Screen)'; this mode boosts the selected signal by using a Screen overlay operation, boosting the signal non-linearly.
- 'Pass'; only lets through the selected signal and attenuates all other signal.

- 'Reject'; blocks the selected signal, leaving all other signal intact.
- 'Fringe Killer'; Draws colour from neighbouring pixels that are not masked and gives these colors to masked pixels. Note that this mode requires a mask to be set.
- 'Saturate Visual H-alpha'; saturates red coloring. In this mode, the user must click on the coloring that is to be *preserved* while the H-alpha is boosted.
- 'Saturate Visual H-beta/O-III'; saturates cyan coloring. In this mode, the user must click on the coloring that is to be *preserved* while the H-beta/O-III is boosted.

The '**Filter Width**' parameter specify the responsiveness of neighbouring colors in the spectrum. A small '**Filter Width**' will see the module only modify areas with a very precise match in colour to the area selected, while a larger '**Filter Width**' will see the module progressively modify areas that deviate in colour from the selected area as well.

The '**Sampling Method**' mode selects how a click on the image samples the image. The '3x3 Average' mode samples a 3x3 area around the clicked pixel and uses the resulting 9-pixel average as the input colour. The 'Single Pixel' mode, samples only the precise pixel that was clicked.




Operating the module is as easy as picking a Filter Mode and clicking on areas you wish to affect with your chosen filter.

Finally, a '**Mask Fuzz**' parameter allows for the result to progressively masked in cases where a mask is set.

## Mitigating chromatic aberration

The Filter module's 'Fringe Killer' mode is an easy and very effective way to remove unsightly blue and purple halos caused by chromatic aberration.

Simply put the offending stars, including their halos in a mask (one can be automatically generated from within the Filter module, by clicking Mask, Auto, Stars or FatStars, Do, Keep). Next click a few times on different parts of the purple or blue halos and they will slowly disappear with each click.

The Filter module's 'Fringe Killer' mode is an easy and very effective way to remove unsightly blue and purple halos caused by chromatic aberration.

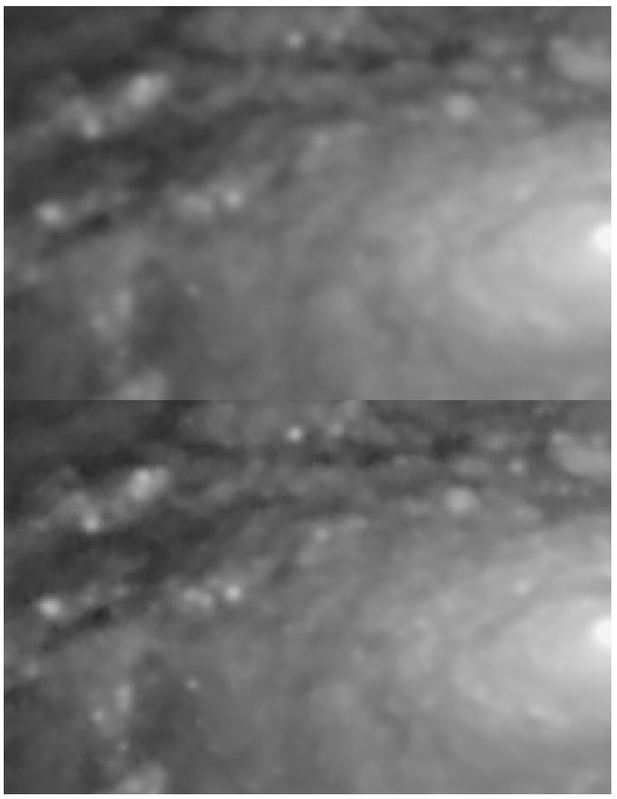
# Flux: Automated Astronomical Feature Recognition and Manipulation

The Fractal Flux module allows for fully automated analysis and subsequent processing of astronomical images of DSOs.

The one-of-a-kind algorithm pin-points features in the image by looking for natural recurring fractal patterns that make up a DSO, such as gas flows and filaments. Once the algorithm has determined where these features are, it then is able to modify or augment them.

Knowing which features probably represent real DSO detail, the Fractal Flux is an effective de-noiser, sharpener (even for noisy images) and detail augmenter.

Detail augmentation through flux prediction can plausibly predict missing detail in seeing-limited data, introducing detail into an image that was not actually recorded but whose presence in the DSO can be inferred from its surroundings and gas flow characteristics. The detail introduced can be regarded as an educated guess.



Flux sharpening by self similarity.

It doesn't stop there however – the Fractal Flux module can use any output from any other module as input for the flux to modulate. You can use, for example, the Fractal Flux module to automatically modulate between a non-deconvolved and deconvolved copy of your image – the Fractal Flux module will know where to apply the deconvolved data and where to refrain from using it.

Flux Sharpening by self-similarity feature detection. Only areas that are deemed recurring detail are sharpened.

# usage

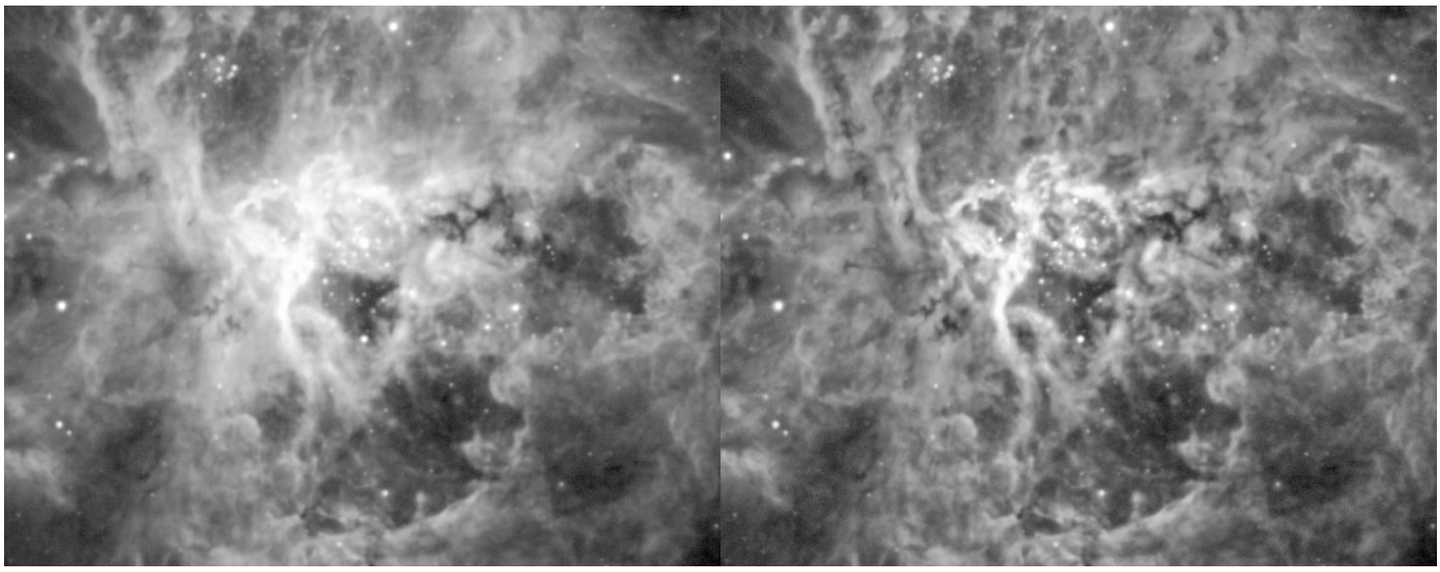
coming soon

# HDR: Automated Local Dynamic Range Optimization

The HDR (High Dynamic Range) module optimises *local* dynamic range, recovering small to medium detail from your image. The module intuitively and effortlessly lets you resolve detail in bright galaxy cores, faint detail in nebulas and works just as well on solar, lunar and planetary images.

This third iteration of the HDR module (as of StarTools 1.8), makes it easy to achieve natural results with minimal (or no) visible artifacts or star bloat, while making full use of the signal evolution Tracking engine.

A HDR optimisation tool is a virtual necessity in - particularly - deep space astrophotography, owing to the huge brightness differences (aka 'dynamic range') innate to various objects that exist in deep space.



The HDR module puts you in full, intuitive control of local dynamic range allocation, without introducing artifacts or making the image look unnatural. Data acquisition by Marc Aragnou.

# usage

The HDR module optimises local dynamic range allocation for small to medium areas than the Contrast module. As such it ideally complements a prior application of the Contrast module.

## At a glance

The HDR module combines multiple strategies/algorithms into one signal flow;

1 Local gamma correction, solves for an "ideal" per-pixel gamma correction by evaluating histogram shape (such as Pearson mode

''skewness'' and other statistical properties) in the context of a pixel's immediate surroundings.

2 Local histogram remapping, solves for the ''ideal'' luminance value per-pixel, based on its place in a local histogram, taking into account maximum spatial(!) contrast values.

3 Signal evolution Tracking-driven noise grain rejection ensures that - normally - noise-prone local histogram equalization (LHE), yields more robust estimates for signal/detail vs noise grain.

The HDR module operates exclusively on the luminance component of your image, retaining any coloring from the input image.

## Launching the HDR module

Depending on the size (X * Y resolution) of the dataset at hand, the once-off initial processing/analysis may take some time, particularly at high resolution datasets and high '**Context Size**' settings. Note that this processing/analysis is repeated every time the **'Context Size'** parameter is changed, or when a new preview area is specified. Processing times may be cut by opting for a lower precision local gamma correction solving stage via the '**Quality**' parameter.

However, once this initial processing/analysis has completed any parameter modification that does not involve **'Context Size'**, will complete virtually in real-time.

## Presets

As with most modules in StarTools, the HDR module comes with a number of universally applicable presets that demonstrate settings for various use case;

- **Reveal**; corresponds to the default settings, and combines moderate local gamma correction for the highlights with moderate local detail enhancement in both the shadows and highlights. This preset (and default setting) tends to be a generally applicable example.
- **Tame**; targets detail recovery in the highlights by applying aggressive local gamma correction in larger highlight areas. This presets demonstrates the HDR module's excellent ability to bring larger areas in the highlights under control and reveal any detail they might contain. This preset is, for example, useful to bring bright galaxy cores under control and reveal their detail.
- **Optimize**; targets and accentuates smaller detail in both shadows and highlights equally.
- **Equalize**; pulls both dim and bright larger contiguous areas into the midtones equally.

## Parameters

Evaluating the effect of the above presets, the intuitive nature of the parameters become clear;

The **'Highlights Detail Boost'** and **'Shadows Detail Boost'** parameters generally provide a means to accentuate existing detail without affecting the brightness of larger contiguous areas, preserving that context.

The **'Gamma Highlights'** and **'Gamma Shadows'** parameters generally provide a great dynamic range management solution for larger contiguous areas that are very bright (or dim), however contain smaller scale detail.

The '**Gamma Smoothen**' parameter controls the smoothness of the transition between differently locally stretched areas. Thought the default value tends to be applicable to most situation, you can increase this value if any clear boundaries can be seen, or you decrease this value to get a clearer idea of which areas are modified (and how).

The **'Signal Flow'** parameter specifies the signal sources for the algorithm stack.;

- **Tracked**; uses a version of the signal that fully takes into account noise grain propagation in the signal. This allows the module to disregard recovered 'detail' in low-SNR areas that can be attributed to stretching the noise component of the signal, rather than the signal itself. Using this setting is highly recommended if you use HDR as part of a larger workflow, and plan on further detail recovery processing, particularly with algorithms like deconvolution.

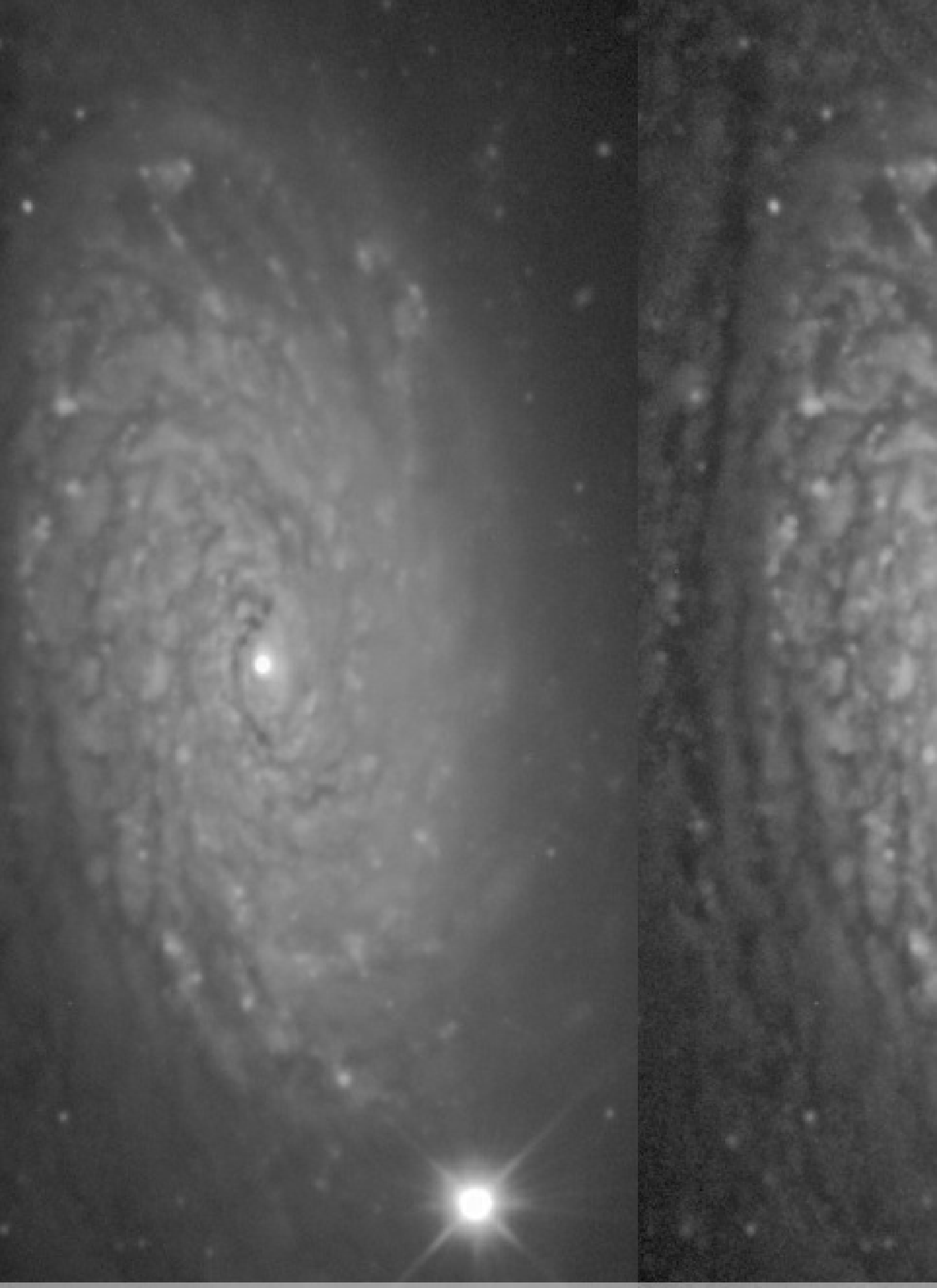
- **Visual As-Is**; uses the stretched image (exactly as visible before launching the HDR module), without further noise propagation compensation.

The **'Context Size'** parameter controls the upper size of the detail/structures that may provide context for smaller detail. For example, reducing this parameter will see increasingly smaller detail being accentuated, with less and less concern for larger detail. A smaller **'Context Size'** value may be appropriate in cases where resolving small detail is of higher priority and larger scale context is ideally ignored (for example globular clusters). The previously mentioned caveats for changing this parameter apply; high values tend to help preserving large scale context well, but may incur longer initial processing times. Processing times may be cut by opting for a lower precision local gamma correction solving stage via the **'Quality'** parameter.

## Artifacts

Results from the HDR module are generally artifact-free, unless using rather extreme values. This third iteration of the module was specifically engineered to further minimise the artifacts of alternative implementations (such as HDRWT and AHE/CLAHE). Star "bloat" or ringing artifacts should be negligible under normal operating conditions, while noise-induced "detail" development is suppressed through the incorporation of signal evolution Tracking statistics. Highlights vs Shadows manipulations are available independently, and applying just one or the other should not yield any detectable sharp transitions.

More caution should be exercised when using extreme values far outside of the defaults or presets.



From left to right; Original, 'Reveal' preset, 'Tame' preset, 'Optimize' preset, 'Equalize' preset. Data acquisition by Jim Misti.

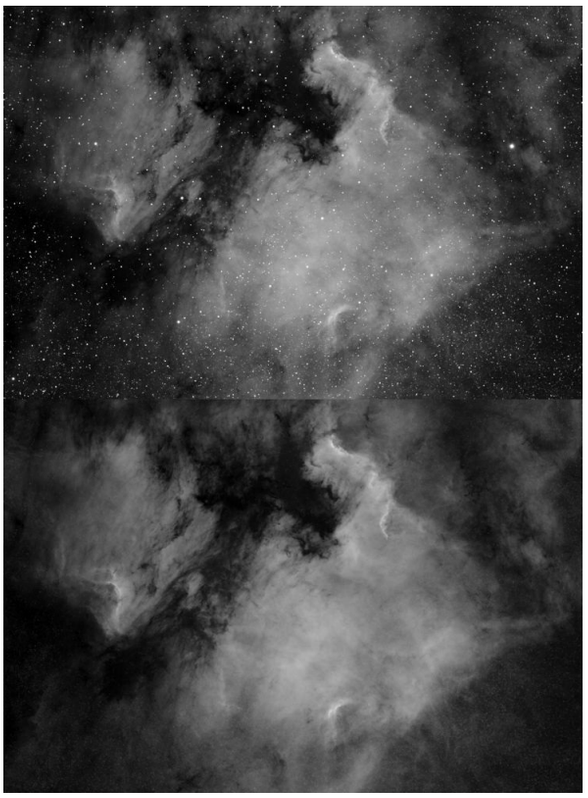
200% zoom image, showing the effects of signal evolution Tracking on image quality. Left: original, Middle: 'Optimize' preset detail recovery with 'Signal Flow' set to 'Visual As-Is', Right: 'Optimize' preset detail recovery with 'Signal Flow' set to 'Tracked'.

# Heal: Unwanted Feature Removal

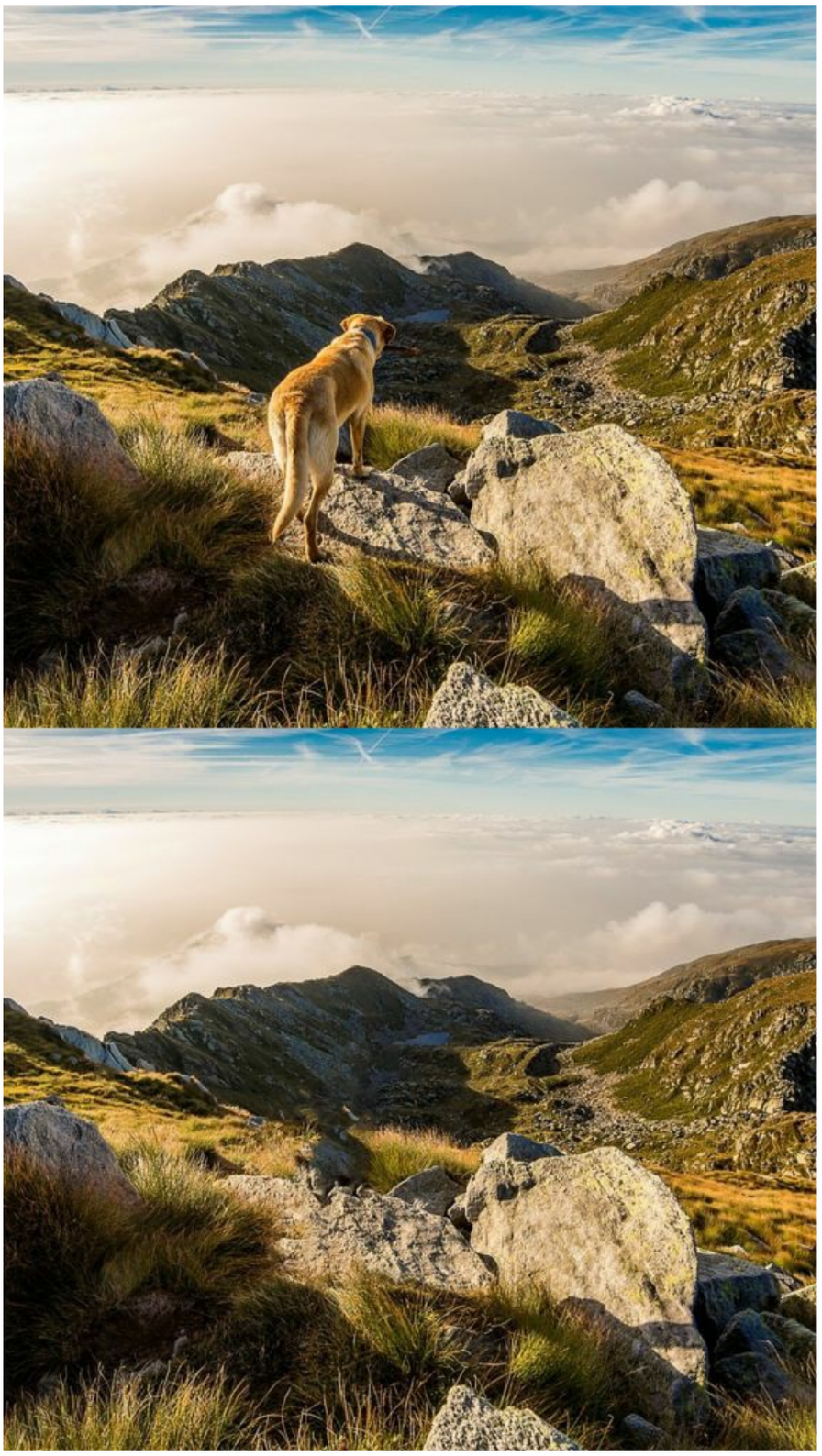
The Heal module was created to provide a means of substituting unwanted pixels in an neutral way.

Cases in which healing pixels may be desirable may include the removal of stars, hot pixels, dead pixels, satellite trails and even dust donuts.

The Heal module incorporates an algorithm that is content aware and is able to synthesise extremely plausible substitution pixels for even the large areas. The algorithm is very similar to that found in the expensive photo editing packages, however it has been specifically optimised for astrophotography purposes.



Removal of stars is an effective way to draw attention to the underlying nebulosity, or to process nebulosity spearately from the stars.



The heal module's algorithm is similar to that found in expensive photo editing packages

# usage

Getting started with the Heal module in StarTools is a fairly straightforward affair; simply put any unwanted pixels in a mask and let the module do its thing. The more pixels are in the mask, the more the Heal module will have to 'invent' and the longer the Heal module will take to produce a result.

By using the advanced parameters, the Heal module can be made useful in a number of advanced scenarios.

The '**New Must Be Darker Than**' parameter lets you specify a brightness value that indicates the maximum brightness a 'new' (healed) pixel may have. This is useful if you are healing out areas that you later wish to replace with brighter objects, for example stars. By ensuring that the 'new' (healed) background is always darker than what you will be placing on top, you can simply use, for example, the Lighten mode in the Layer module.

The '**Grow Mask**' parameter is a quick way of temporarily growing the mask (see Grow button in the Mask editor). This is useful if your current mask did not quite get all pixels that needed removing.

The '**Quality**' parameter influences how long the Heal module may look for substitutes for each pixel. Higher quality settings give marginally better results but are slower.

The '**Neighbourhood Area**' parameter sets the size of the local area where the algorithm can look for good candidate seed pixels.

The '**Neighbourhood Samples**' parameter is useful if you are looking to generate more 'interesting' areas, based on other parts of the image. It can be useful for a large area being healed to avoid small repeating patterns. This feature is useful for terrestrial photography, however, this is often not needed or desirable for astrophotographical images. If you do not wish to use this feature, keep this value at 0.

The '**New Darker Than Old**' parameter sets whether newly created pixels should always be darker than the old pixels. This may be useful for manipulation of the image in the Layer module (for example subtracting the healed image from the original image).

## Using the Heal module with StarNet++

This guide lets you create starless *linear* data using StarNet++ and the Heal module. Even if you wish to use StarNet++ on your final image, you will find that using this guide to extract the a starmask, the Heal will achieve superior results when removing the stars that StarNet++ identified.

Link to forum item about tutorial

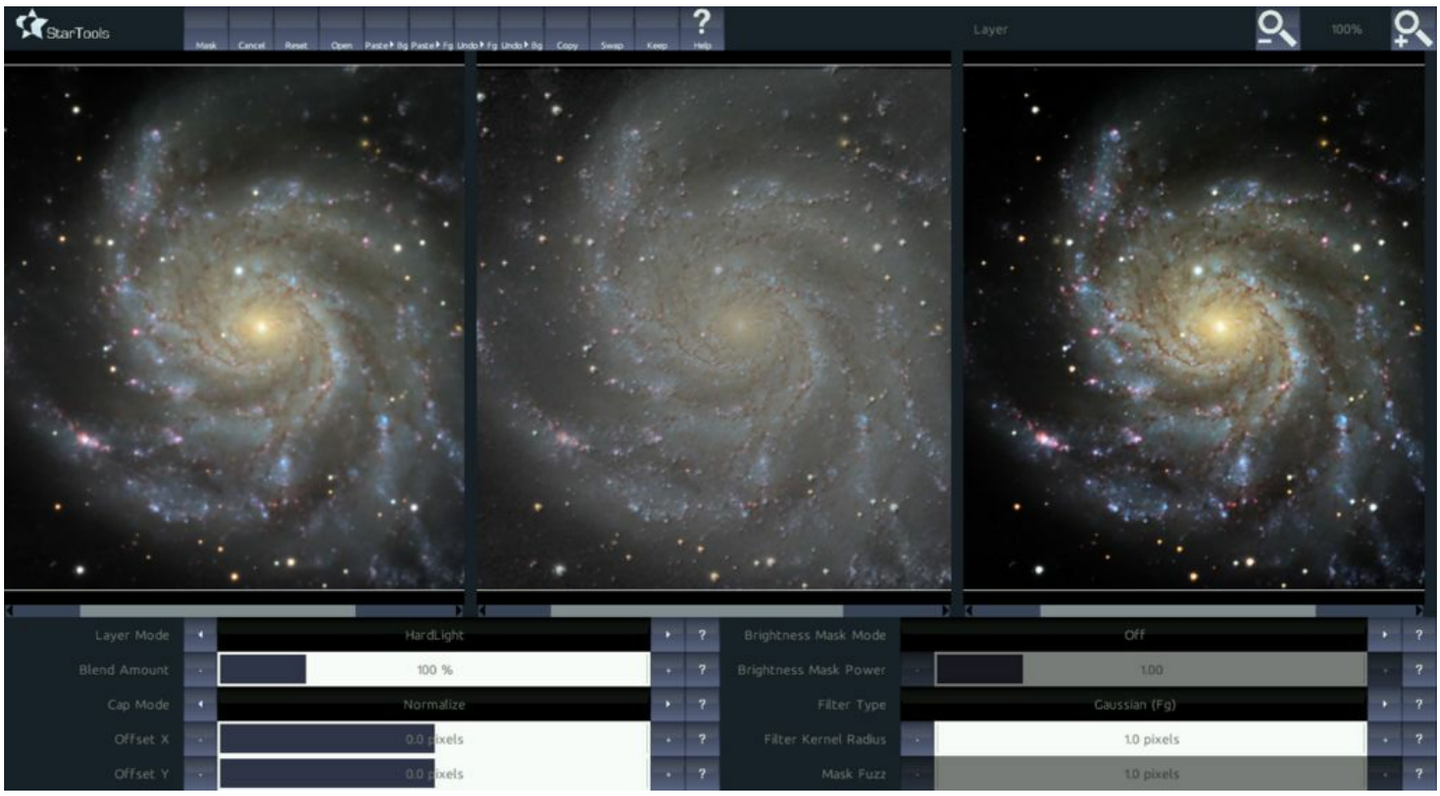
http://forum.startools.org/viewtopic.php?f=7&t=1476

# Layer: Versatile Pixel Workbench

The Layer module is an extremely flexible pixel workbench for advanced image manipulation and pixel math, complementing StarTools' other modules.

It was created to provide you with a nearly unlimited arsenal of implicit functionality by combining, chaining and modulating different versions of the same image in new ways.

Features like selective layering, automated luminance masking, a vast array of filters (including Gaussian, Median, Mean of Median, Offset, Fractional Differentation and many, many more) allow you to emulate complex algorithms such as SMI (Screen Mask Invert), PIP (Power of Inverse Pixels), star rounding, halo reduction, chromatic aberration removal, HDR integration, local histogram optimization or equalization, many types of noise reduction algorithms and much, much more.




The Layer module allows you to chain, mask and layer and apply countless of operations and filters.

# usage

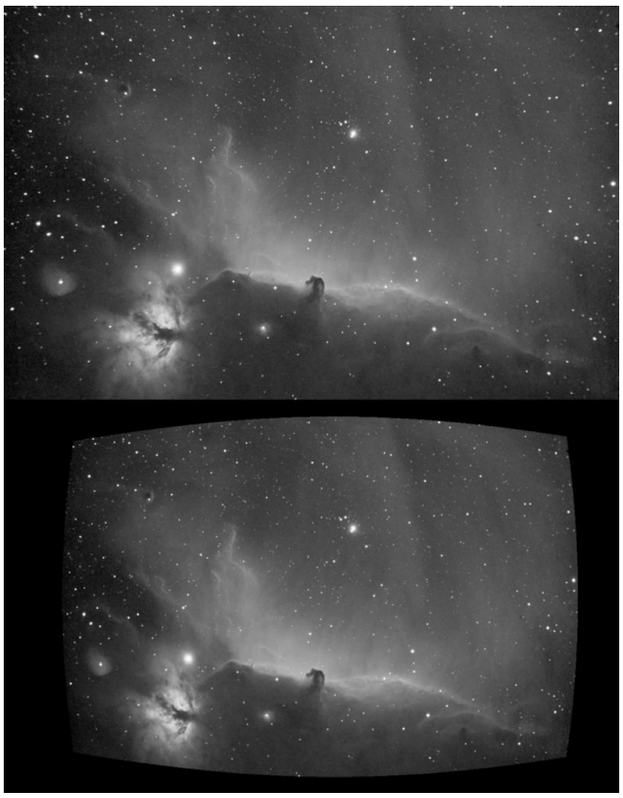
coming soon

# Lens: Distortion Correction and Field Flattening

The Lens module was created to digitally correct for lens distortions and some types of chromatic aberration in the more affordable lens systems, mirror systems and eyepieces.

One of the many uses of this module is to digitally emulate some aspects of a field flattener for those who are imaging without a physical field flattener.

While imaging with a hardware solution to this type of aberration is always preferable, the Lens module can achieve some very good results in cases where the distortion can be well modeled.



Top: source image (courtesy of Marc Aragnou), notice star elongation towards corners. Bottom: Lens corrected image (without auto crop to show curvature).

# usage

coming soon

# NBAccent: Adding Narrowband Accents to Visual Spectrum Datasets

Adding narrowband accents to visual spectrum datasets has traditionally been a daunting, difficult and laborious process, involving multiple workflows. The NBAccent module is a powerful module that starts its work as soon as your load your data in the Compose module. Crucially it adds only a *single*, easy step to an otherwise standard workflow, while yielding superior results in terms of color fidelity/preservation.

By making narrowband accents an integral part of the complete workflow and signal path, results are replicable, predictable and fully tracked by StarTools' unique signal evolution Tracking engine, yielding perfect noise reduction every time.

The NBAccent module adds narrowband accents to visual spectrum datasets. Visibility of HII areas in this image of M33 is greatly enhanced, while visual spectrum coloring is largely maintained. Top: before NBAccent module, Bottom: after NBAccent module.

# Usage

Activating the NBAccent module functionality, starts with importing a suitable narrowband dataset via the Compose module. The Compose module will extract the relevant channels from the dataset you give it, as directed by its '**NB Accents Type**' parameter.

The narrowband dataset is processed in parallel during your workflow; the Bin, Crop, Mirror, Rotate and - most notably - Wipe modules all operate on the narrowband accent dataset in parallel as you process the main luminance (and optionally chrominance) signal.

## Understanding the module's purpose and use case

There are many different ways and techniques of incorporating narrowband data into your workflow. Which method is suitable or desirable, depends on the object, the availability of datasets/bands, and the quality of those available dataset.

The NBAccent module was specifically designed for the most difficult compositing use case; that of using narrowband as means to *accentuate* detail in a visual spectrum 'master' dataset. In other words, in this use case, the narrowband is used to support, enhance and accentuate small(er) aspects of the final image, rather than as a basis for the initial signal luminance/detail or chrominance/coloring itself. This is a subtle, but tremendously important and consequential distinction.

As such, the narrowband accent dataset is processed entirely independent of the luminance and chrominance signal of the 'master' dataset; it sole purpose is to accentuate detail from the 'master' (luminance/chrominance) dataset through careful - but deliberate - local brightness and/or color manipulation.

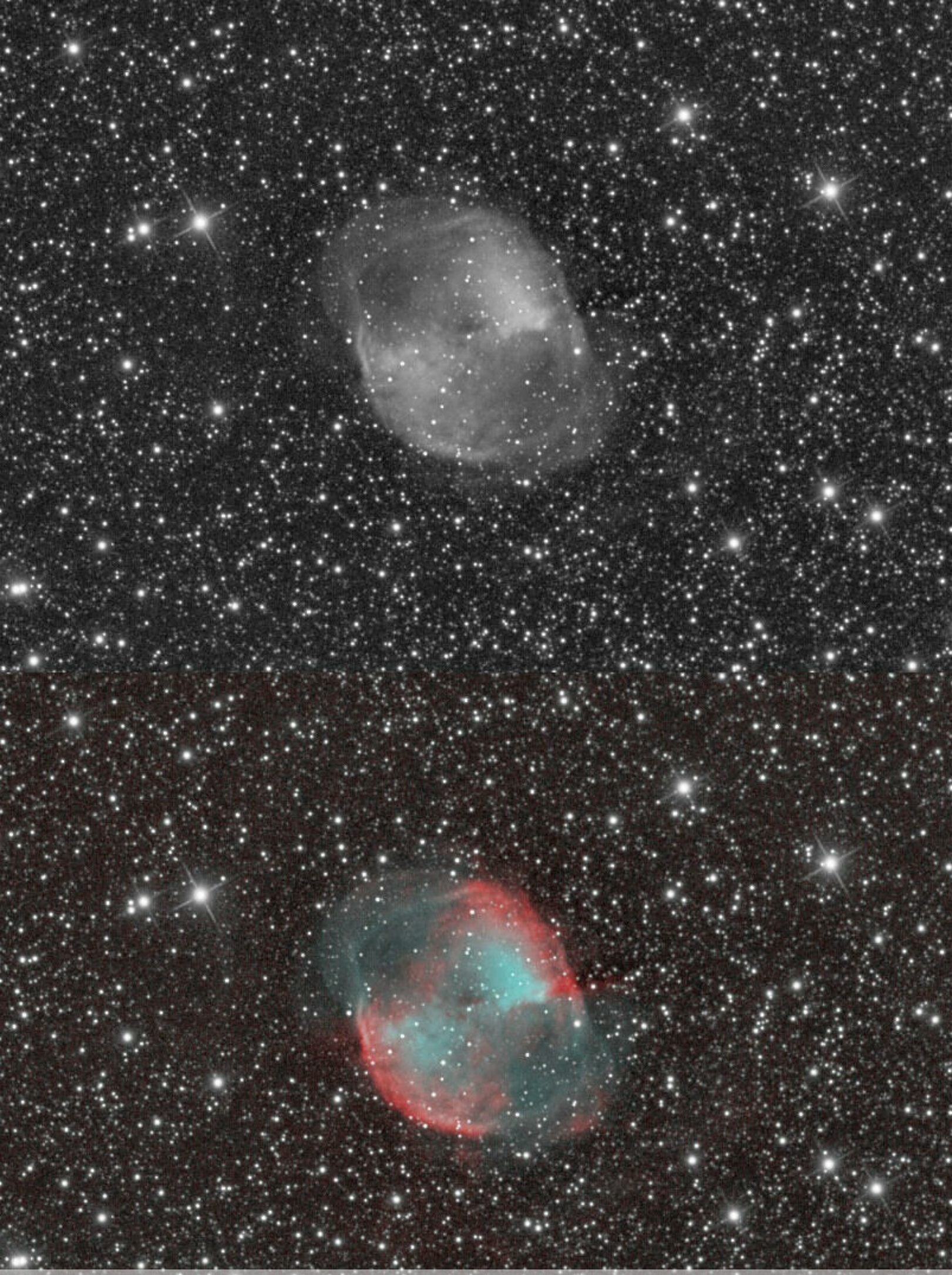
If you wish to use the narrowband signal as luminance or chrominance *itself*, rather than for *accentuating* luminance or chrominance, then the NBAccent module will not apply, and you should use the Compose module to load your narrowband as luminance and/or chrominance instead.

Given the module's use case, it is best invoked late in the processing flow, after the Color module.

Examples of use cases for the NBAccent module are;

- accentuating HII areas in galaxies (by passing it a Hydrogen-alpha dataset) such as M31, M33
- accentuating or adding large scale background nebulosity to already rich visual spectrum widefield renditions of HII areas such as NGC 7635, M16
- accentuating or better resolving intricate features in objects such as planetary nebula

Ideal datasets for augmenting visual spectrum (mono or colour) datasets are Ha datasets, O-III datasets, Ha+O-III datasets or datasets from the popular duo/tri/quadband filters for OSCs and DSLRs such as the Optolong L-Extreme, the STC Duo , the ZWO Duo-Band and other similar filters with narrow spectrum responses.



Through the power of StarTools' three-way signal separation (luminance, chrominance, and narrowband accents), the module can als be deployed for more esoteric uses. In this example it is used to endow a luminance ("L") dataset with narrowband accents acquired through a popular duoband filter. No chrominance (color) dataset was used here. Notice that, as a result, stars remain perfectly white

## Stage 1: Signal stretch and contribution calibration

The first screen allows you to fine control which areas will receive narrowband enhancement. The procedure and, hence, interface is closely related to the AutoDev module. Familiarizing with AutoDev is key to achieving good results with StarTools, and being able to use it effectively is a prerequisite to being able to use the NBAccent module.

One notable difference compared to AutoDev, is the way the stretched narrowband data is presented; areas that *will not* be considered for the final composite, will be clipped to black. Areas that *will* be considered in the final composite, will appear sretched as normal. The other difference from the AutoDev module, is the removal of the 'Detector Gamma' parameter and its replacement by the '**Threshold**' parameter; this parameter allows for intentional clipping of the narrowband image, for example to avoid any background imperfections being added to the final composite. It is important to note that **this parameter should be used as a last resort only** (for example if the narrowband accent data is of exceedingly poor quality) as it is a very crude tool that will inevitably destroy faint signal.

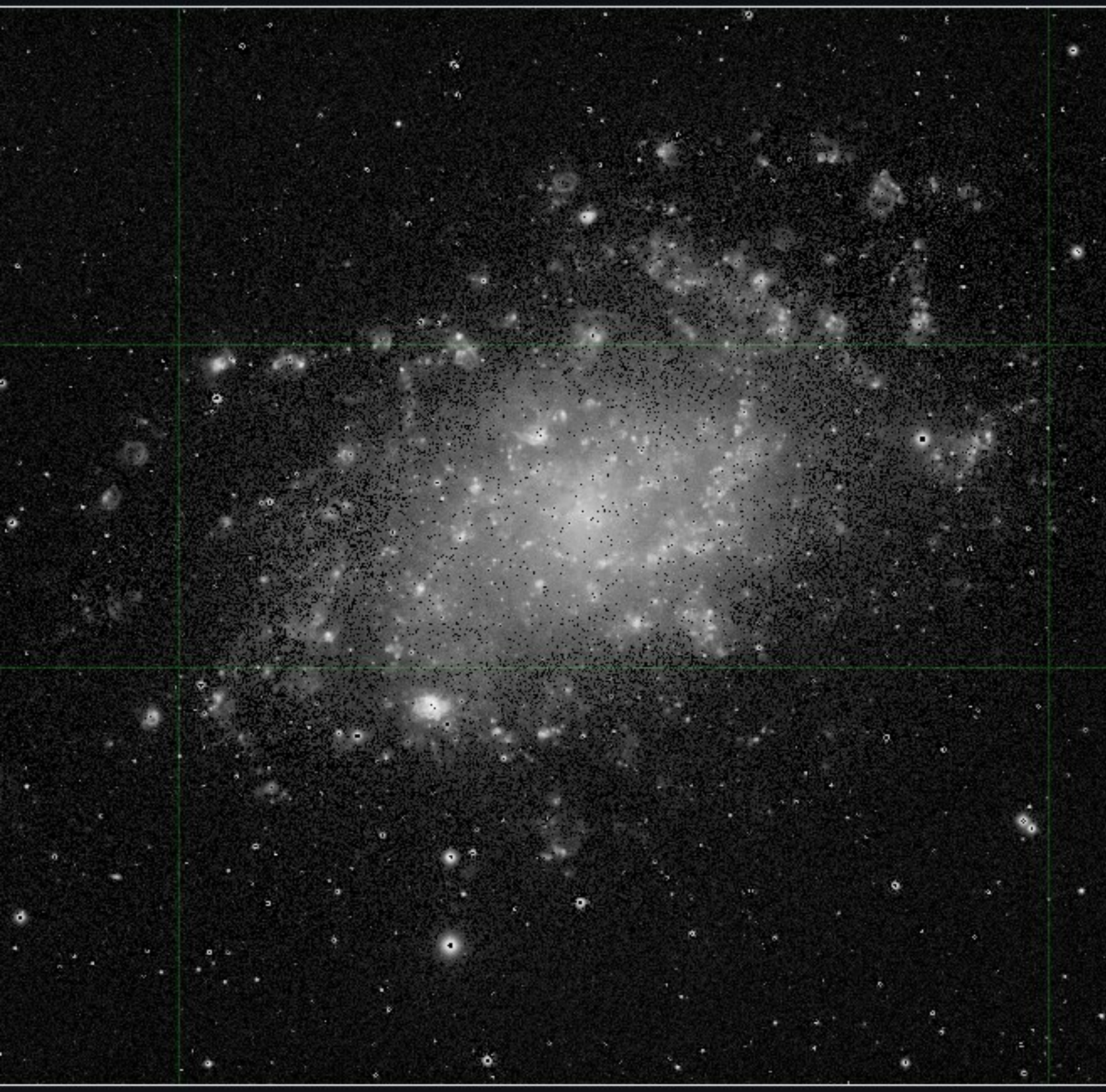
It is important to understand that the signal as show during this first stage, is merely signal that is up *for consideration* by the second stage. Its inclusion is still contingent on other parameters and filters in the second stage. In other words, during this first stage, you should merely ensure that, whichever signal is visible, is actual useful narrowband signal, and

not the result of background imperfections or other artificial sources.

For your convenience, the NBAccent module will, by default, use the same Region of Interest that was specified during AutoDev.

Next | Reset | No RoI | ?
Help

Visual Spectrum Narrowband Accent Compositing



A Region Of Interest (RoI) is active. You can click 'No RoI' to reset RoI.

| | | | | | |
|---|---|---|---|---|---|
| 603 pixels | + | ? | RoI Y1 | - | |
| 2321 pixels (-359) | + | ? | RoI Y2 | - | 130 |
| 4.3 pixels | + | ? | Outside RoI Influence | - | |

In this setup stage, pixels that will be affected in the visual spectrum image will show narrowband signal, while pixels that will not be affected are clipped to black. This will allow you to gauge how the image will be transformed by the parameters you choose here.

## Stage 2: Accentuating your image with narrowband accents

The second stage is all about using the signal from the first stage in a manner you find aesthetically pleasing.

Straight up, there are two presets that are useful in two of the NBAccent's major use cases;

- **'Nebula'**; to accentuate detail associated with Milkyway nebulosity
- **'Galaxy'**; to accentuate smaller detail in other galaxies

These presets dial in the most useful settings for these two usecases.

The **'Response Simulation'** parameter is responsible for the visual spectrum coloring equivalent that is synthesised from the narrowband data. The NBAccent module was designed to synthesise plausible visual spectrum coloring for a wide range of scenarios and filters;

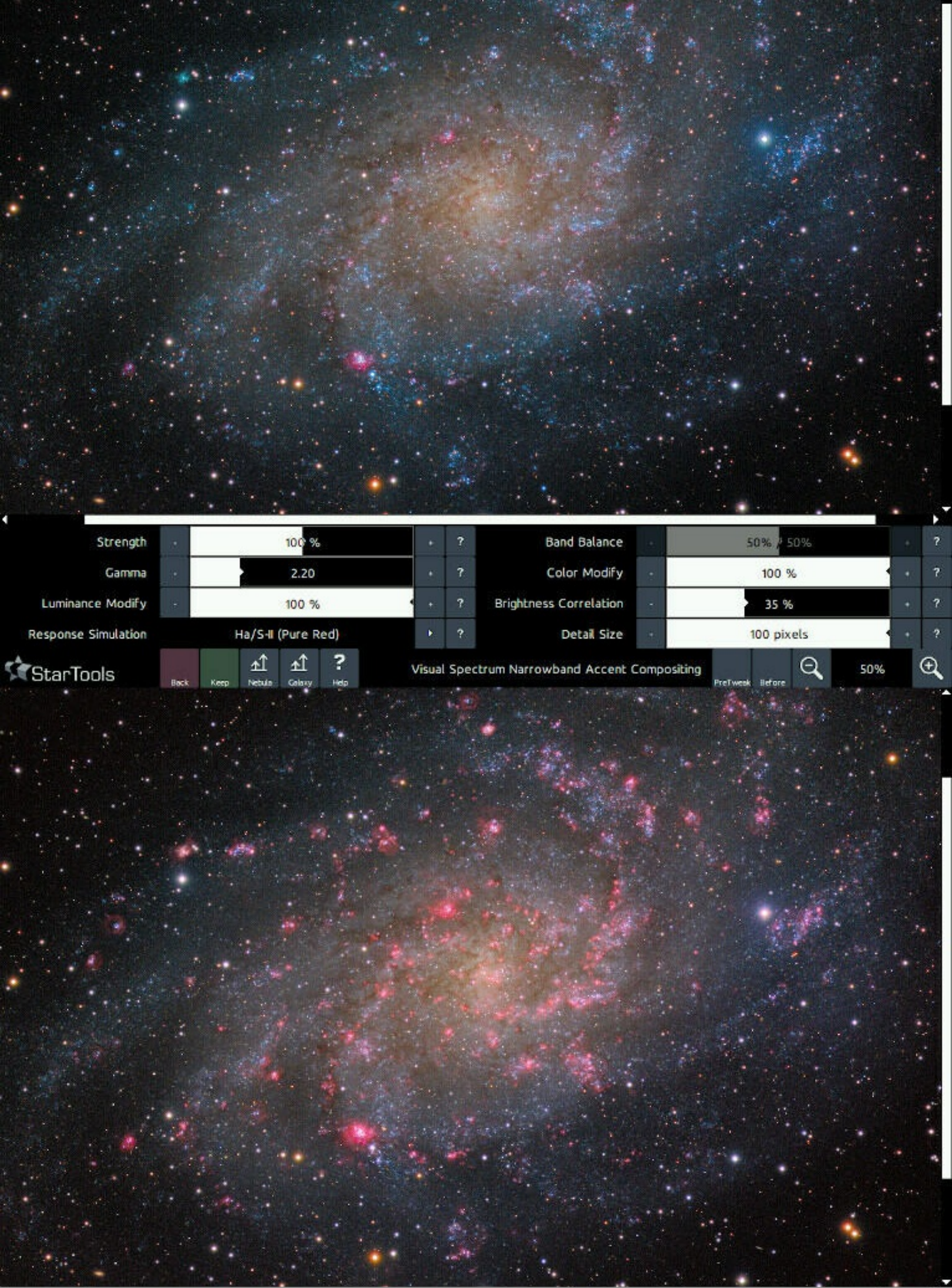
- **Ha/S-II (Pure Red)**; uses the narrowband data's red channel to add pure, deep red accents to the image. While pure red is rather rare in visual spectrum images (due to these emissions almost never existing by themselves and instead being accompanied by other emissions that are much bluer), it can nevertheless be useful to make these areas stand out very well.

- **HII/Balmer Series (Red/Purple)**; uses the narrowband data's red channel to add the familiar red/purple colour of HII areas to the image. This mode makes the assumption that the other visual spectrum emissions from the Balmer series (almost all blue) are also present where the H-alpha line was detected. This mode tends to yield renditions that matches closely with the colouring of HII areas in actual visual spectrum data.
- **Hb/O-III (Cyan)**; uses the narrowband data's green and blue channels to add pure cyan accents, corresponding to the colour of areas of strong Hb/O-III emissions as powered by nearby O or B-class blue giant stars.
- **O-III (Teal)**; uses the narrowband data's green and blue channels to add teal green accents, corresponding to the colour of areas of strong O-III emissions
- **Ha/S-II (Pure Red) + Hb/O-III (Cyan)**; uses pure deep red accents for data from the red channel, while using cyan accents for data from the blue and green channels. This mode is particularly useful for narrowband data acquired through the popular duo/tri/quadband filters.
- **Ha/S-II (Pure Red) + O-III (Teal)**; uses pure deep red accents for data from the red channel, while using teal green accents for data from the blue and green channels. This mode is particularly useful for narrowband data acquired through the popular duo/tri/quadband

filters.

- **HII/Balmer Series (Red/Purple) + Hb/O-III (Cyan)**; synthesises the full Balmer series (red/purple) from the red channel, while using cyan accents for data from the blue and green channels. This mode is particularly useful for narrowband data acquired through the popular duo/tri/quadband filters.
- **HII/Balmer Series (Red/Purple) + O-III (Teal)**; synthesises the full Balmer series (red/purple) from the red channel, while using green accents for data from the blue and green channels. This mode is particularly useful for narrowband data acquired through the popular duo/tri/quadband filters.

The '**Luminance Modify**' and '**Color Modify**' parameters, precisely control how much the module is allowed to modify of the visual spectrum image's luminance/detail and colour respectively. For example, by setting '**Luminance Modify**' to 0%, and leaving '**Color Modify**' at 100%, only the colouring will be modified, but the narrowband accent data will not (perceptually) influence the brightness of any pixels of the final image. Conversely, by setting '**Color Modify**' to 0% and '**Luminance Modify**' to 100%, the narrowband accent data will significantly brighten the image in areas of strong narrowband emissions, however the colouring will remain (perceptually) the same as the visual spectrum input image.




Isolated detail from the first stage is now modulated, color mapped and added to the visual spectrum image. Top: original image ("Before" switch toggled), bottom: image with added Hydrogen-alpha accents mapped to pure red.

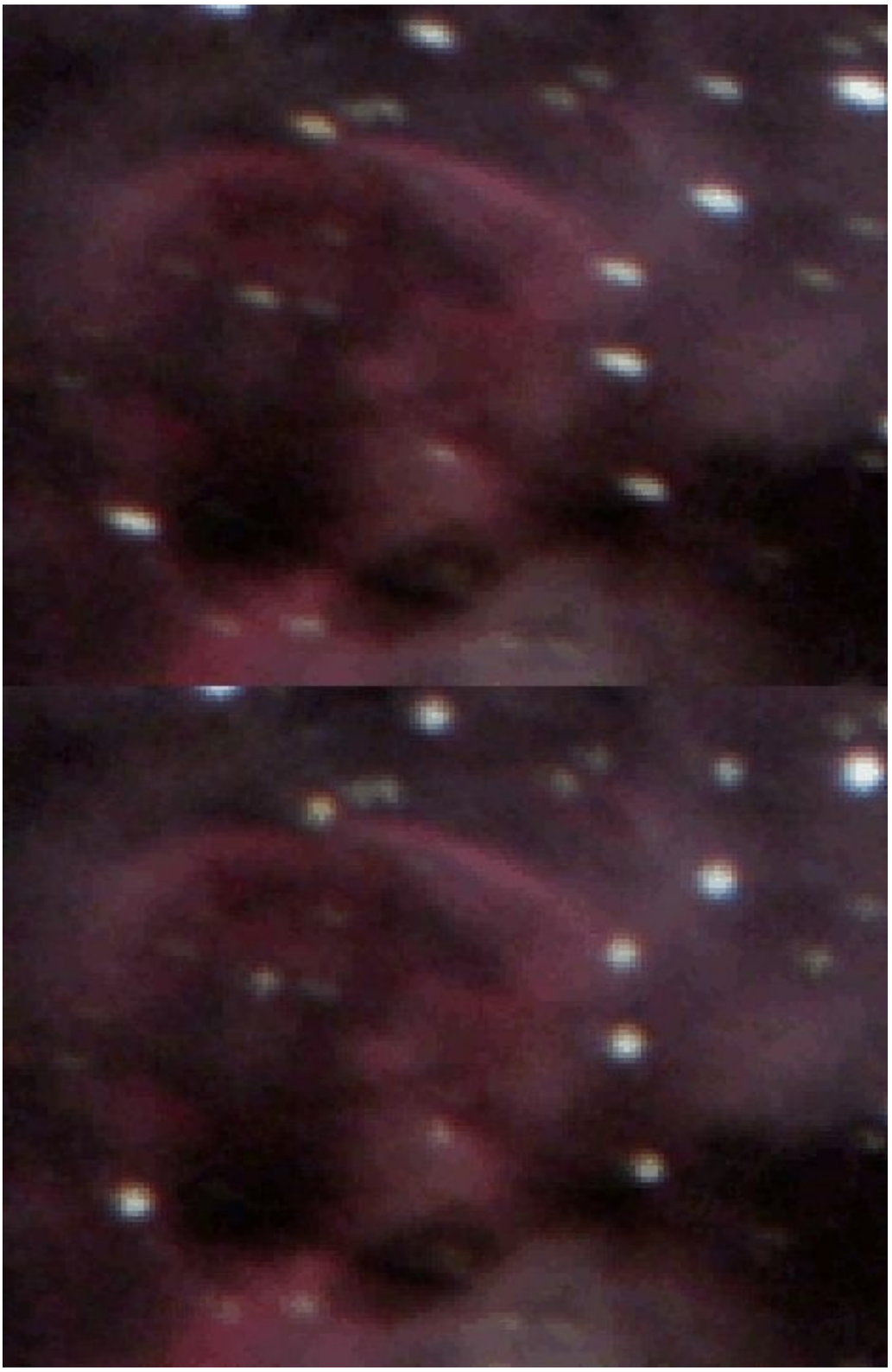
# Repair: Star Rounding and Repair

The Repair module attempts to detect and automatically repair stars that have been affected by optical or guiding aberrations.

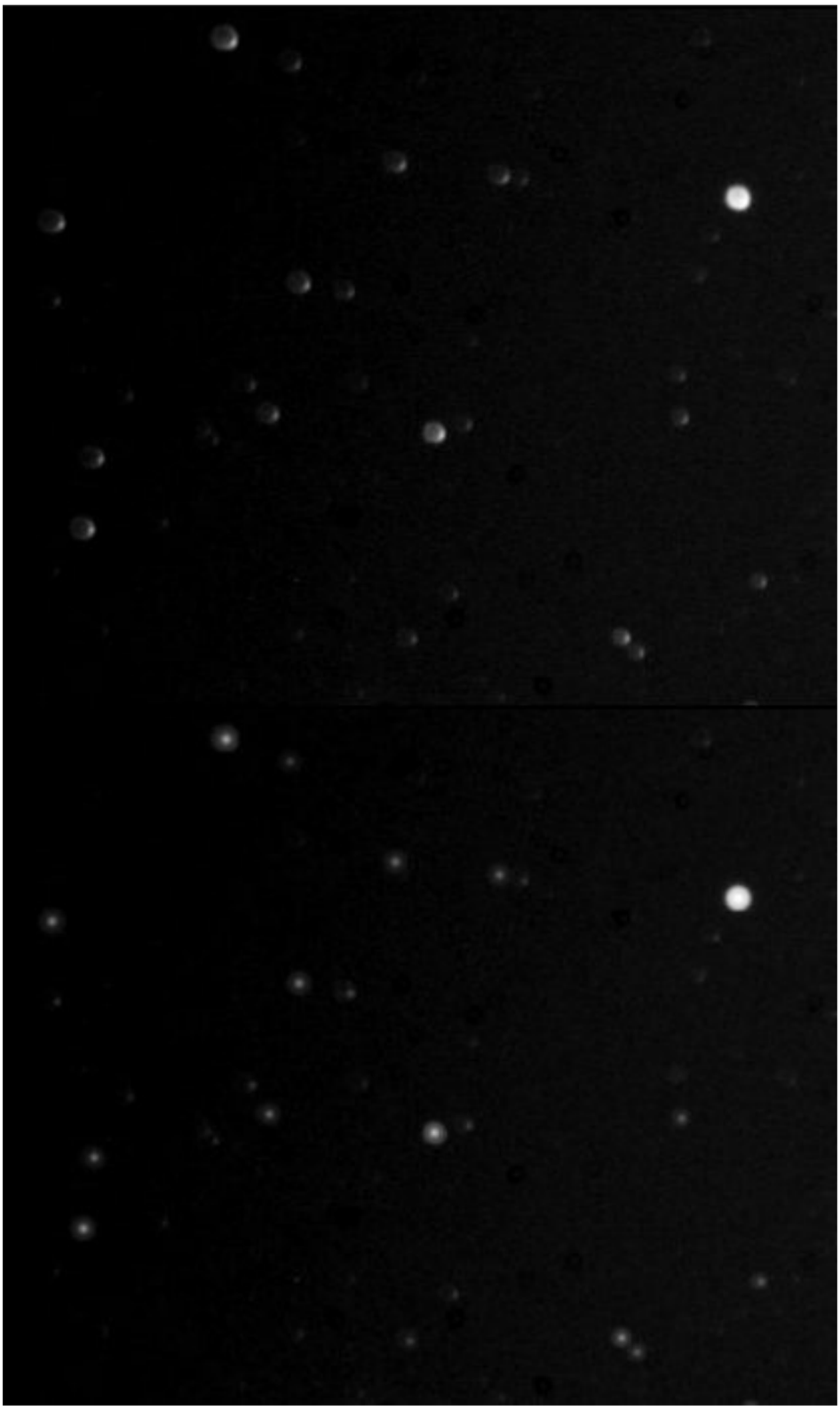
Repair is useful to correct the appearance of stars which have been adversely affected by guiding errors, incorrect polar alignment, coma, collimation issues or mirror defects such as astigmatism.

The Repair module allows for the correction of more complex aberrations than the much less sophisticated 'offset filter & darken layer' method, whilst retaining the star's exact appearance and color.

The repair module comes with two different algorithms. The 'Warp' algorithm uses all pixels that make up a star and warps them into a circular shape. This algorithm is very effective on stars that are oval or otherwise have a convex shape. The 'Redistribution' algoirthm uses all pixels that make up a star and redistributes them in such a way that the original star is reconstructed. This algorithm is very effective on stars that are concave and can not be repaired using the 'Warp' algorithm.



The Repair module's "Warp" algorithm uses the original pixels from the image to reverse-warp stars back into shape.



The Repair module's "Redistribute" algorithm uses the original pixels from the image and recalculates their appearance and position as if they originated from a point light source.

# usage

coming soon

# Sharp: Multi-scale Noise-Aware Structural Detail Enhancement

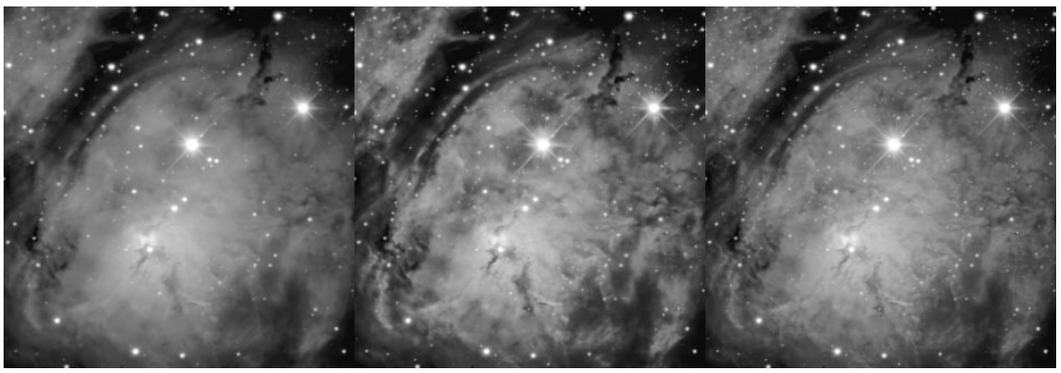
Through wavelet decomposition function *specifically* designed for astrophotographical optical systems, StarTools' Detail-aware Wavelet Sharpening allows you to bring out faint structural detail in your images.

An important innovation over other, less sophisticated implementations, is that StarTools' Wavelet Sharpening gives you precise control over how detail across different scales and SNR areas interact. This means that;

- Sharp lets you control how detail is enhanced, based on the Signal-to-Noise Ratio (SNR) per-pixel in your image. This ability lets you dig out larger scale faint detail entirely without increasing perceived noise.
- Sharp lets you to be the arbiter when two scales (bands) are competing to enhance detail in their band for the same pixel.
- As opposed to other, less desirable implementations (such as median-based wavelet transforms found in some other software), the Sharp module retains all the benefits of a Gaussian transform (e.g. closely resembling the ideal signal responses for detail and PSFs for astrophotographical optical systems) while still avoiding the ringing artifacts. The Sharp module truly combines the best of both worlds.

As with all modules in StarTools, the Wavelet Sharpening module will never allow you to clip your data, always yielding useful results, no matter how outrageous the values you choose, while availing of the Tracking feature's data mining. The latter makes sure that, contrary to other implementations, only detail that has sufficient signal is emphasised, while noise grain propagation is kept to a minimum.

The Sharp is able to dig out faint detail in the form of larger structures, entirely without exacerbating noise.



Left: source. Middle: bias towards larger scale structures. Right: bias towards smaller scale structures.

# Usage

Using the Sharp module, starts with specifying an upper limit of the size of the detail that should be accentuated via the '**Structure Size**' parameter. You should only need to change this parameter if you wish very fine control over small details.

After pressing 'Next', a star mask should be created that protects bright stars (and their extending profiles) from being accentuated.

An '**Amount**' parameter governs the strength of the overall sharpening.

The '**Scale *n***' parameters allow you to control which detail sizes are getting enhanced. If you wish to keep small details from being enhanced set '**Scale 1**' to 0%, similarly if you wish to keep the very largest structures from being enhanced, set '**Scale 5**' to 0%.

The '**Dark/Light Enhance**' parameter gives you control over whether only bright or dark (or both) detail should be introduced.

The two '**Size Bias**' parameters controls the detail size that should prevail if two scales are 'fighting' over enhancing the same pixel. A higher value gives more priority to finer detail, whereas a lower value gives more priority to larger scale structures. It is this ability of the Sharp modules, to dynamically switch between large and small detail enhancement that makes every combination of settings look coherent without 'overcooking' the image; the adage is that if you try to make everything

(every scale) stand out, nothing stands out. And this is precisely what the Sharp module was designed for to avoid. Inherent to this approach is also the lack of ringing artefacts around sharp edges, even though the module does not employ a (less-ideal) multi-scale median transform to try to circumvent this. This combines the benefits of the response of a pure Gaussian transform (such as precise band delineation in an astrophotographical optical train, as well as noise modelling) with ringing artefact-free detail enhancement.

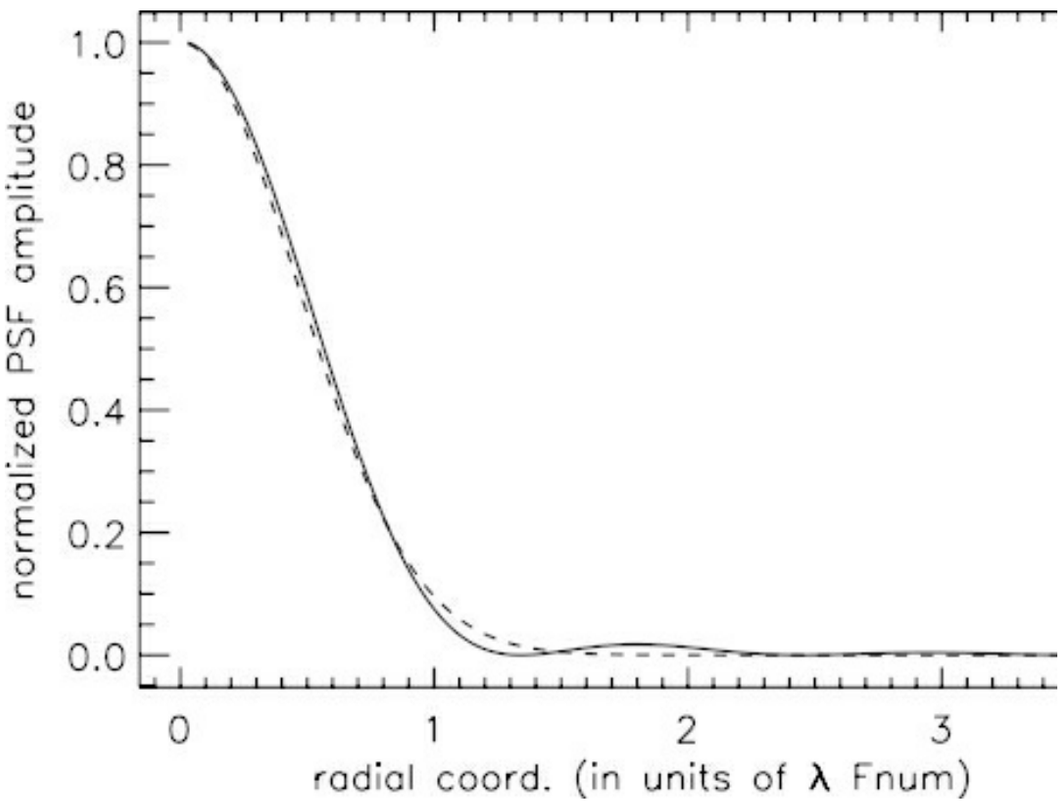
Two version of the '**Size Bias**' parameter exist; the '**High SNR Size Bias**' parameter and the '**Low SNR Size Bias**' parameter. The distinction lies in a further refinement of where and how detail enhancement should be applied. The '**High SNR Size Bias**' parameter controls the size priority for areas with a high signal-to-noise ratio (good signal), whereas the '**Low SNR Size Bias**' controls the size priority for areas with a low signal-to-noise ratio (poor signal).

When Tracking is on, the Tracking feature tends help the Sharp module do a very precise job in making sure that noise is not exacerbated - you may find that the distinction is not needed for most datasets with signal of reasonable quality. However when Tracking is off, these parameters use local luminosity as a proxy for signal quality and the distinction between Low and High SNR will be much more important.

Finally, the '**Mask Fuzz**' parameter increasingly smoothens the area over which the set mask goes from fully in effect, to not in effect.

## Masked vs unmasked areas

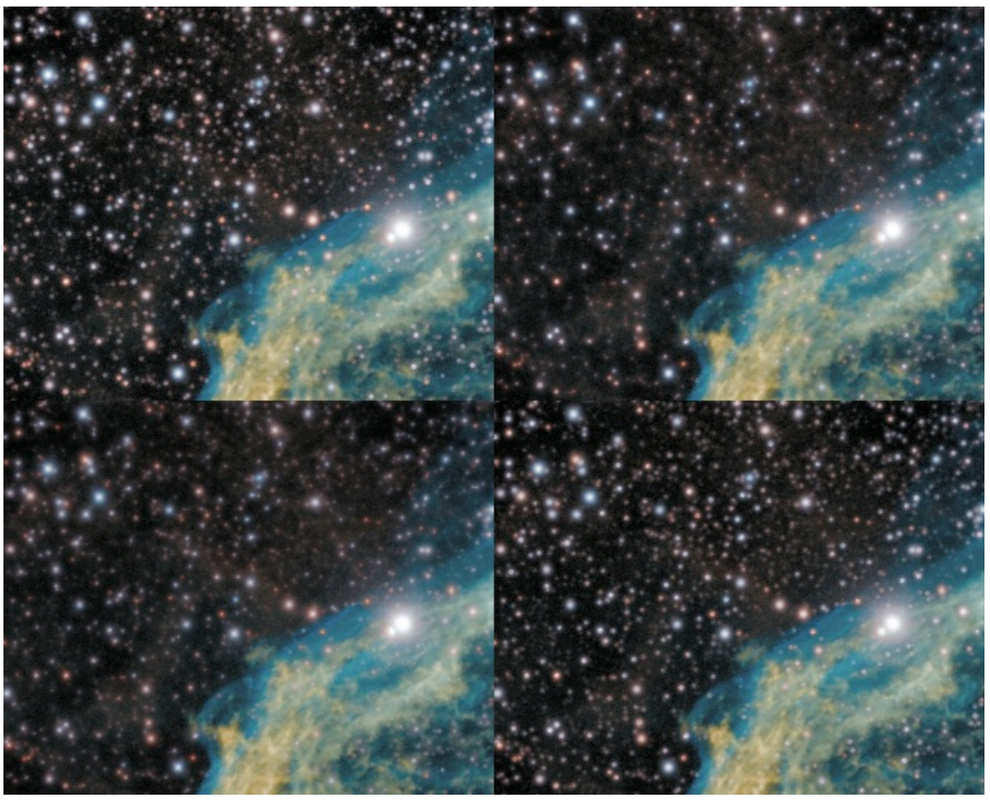
Masks in the Sharp module are primarily used to indicate to the module where stars - and their halos - are located. However, even when masked out, these areas still get processed, though in a subtly different way; only dark detail is emphasised, but not light detail. This avoids accentuating of halos and star "bloating", yet still digs out detail that a stellar halo might be obscuring.




The Sharp module was designed specifically to model detail in astronomical optical systems. This was done by relying on a Gaussian approximation for how detail is diffracted by both Airy disc and any atmospheric turbulence. This graph shows a radial cross-section through the Airy pattern (solid curve) and its Gaussian profile approximation (dashed curve).

# Shrink: Star Appearance Manipulation

The Shrink module offers comprehensive stellar profile modification by shrinking, tightening and re-colouring stars.



Top left: input image. Top right: Tighten preset. Bottom left: Dim preset. Bottom right: Un-glow preset with all else turned off; a subtle contrast increase can be seen around bright stars.

## Usage

A good star mask is essential for good results. Even though the Shrink module is much more gentle on structural detail, ideally, only stars are treated and not any structural detail.

The 'AutoMask' button launches a popup with access to two quick ways of creating a star mask. This same popup is shown upon first launch of the module. The generated masks tend to catch all major stars with very few false positives. If you also wish to include fainter, small stars in the mask, then more sophisticated techniques are recommended to avoid including other detail.

Finally, if your object is mostly obscured by a busy star field, for example in a widefied, then also consider using the Super Structure module to enhance the super structures in your image and push back the busy star field. Combining both the Shrink module's output and the Super Structure module's output can greatly transform a busy looking image in positive ways.

## Parameters

Two '**Mode**' settings are available;

- 'Tighten' has the effect of tightening a stars around their central core.
- 'Dim' has the effect of dimming stars luminosity.

The Shrink module uses an iterative process; the strength of the Tighten or Dim effect is controlled by the number of '**Iterations**', as well as the

'**Regularization**' parameter that dampens the effect. The stringing and pitting artefacts commonly produced by less sophisticated techniques, is thereby avoided.

The '**Color Taming**' parameter forces stars to progressively adopt the colouring of their surroundings, like ''chameleons''.

The '**Halo Extend**' parameter effectively grows the given mask temporarily, thereby including more of each star's surroundings.

If the image has been deconvolved or sharpened and the stars may be subject to subtle ringing artefacts, then the '**De-ringing**' parameter will take this into account when shrinking the stellar profiles, as to not exacerbate the ringing.

The 'Un-glow' feature attempts to reduce the halos around bright, over-exposing stars. '**Un-glow Strength**' throttles the strength of the effect. The '**Un-glow Kernel**' specifies the width of the halos.

Top: original image. Bottom: 'Color Taming' parameter used. Note the stars now appear less conspicuous as their colours blend in with the rest of the object.

The Shrink module's iterative algorithm avoids these sort of stringing and pitting artefacts typically produced by unsophisticated morphological transformations.

## Creating a suitable star mask

A good star mask is essential for good results. Though the Shrink module is much more gentle on structural detail than the basic unsophisticated morphological transformations (such as minimum filters) found in other software, ideally, only stars are treated and not any nebulosity, gaseous filaments or other structural detail.

The 'AutoMask' button launches a pop-up with access to two quick ways of creating a star mask. This same popup is shown upon first launch of the module. The generated masks tend to catch all major stars with very few false positives. If you also wish to include fainter, small stars in the mask, then more sophisticated techniques are recommended to avoid including other detail.

Besides touching up the mask by hand, it is also possible to combine the results of an aggressive auto-generated star mask (catching all faint stars), with a less aggressive auto-generated star mask (catching fewer faint stars, but also leaving structural detail alone);

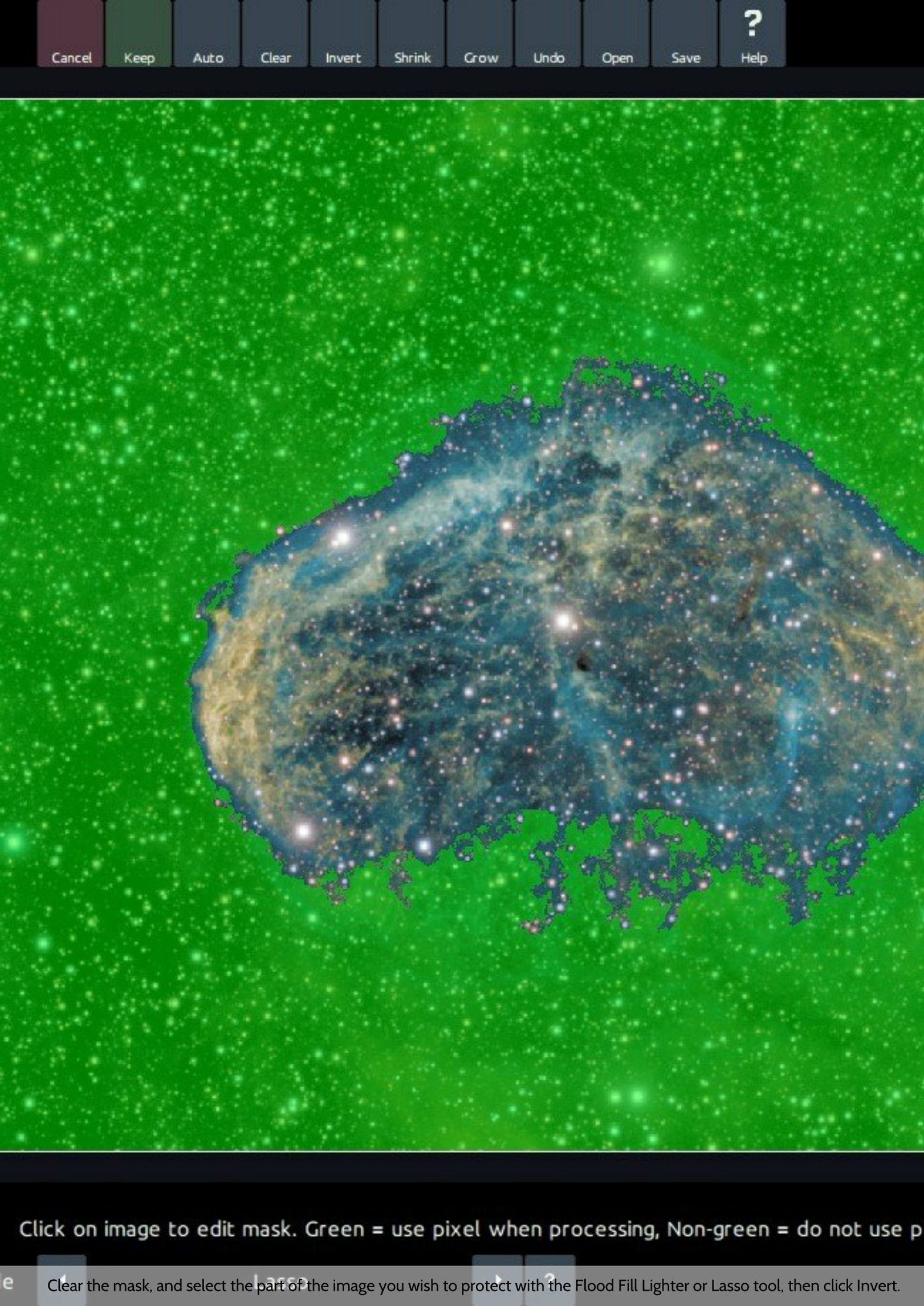
1 Clear the mask, and select the part of the image you wish to protect with the Flood Fill Lighter or Lasso tool, then click Invert.

2 In the Auto mask generator, set the parameters you need to generate your mask (here we choose the 'Stars' preset and set the '**Source**' parameter to 'Stretched' to avoid any noise mitigation measures that may otherwise filter out faint stars for selection). Be sure to set '**Old**

**Mask**' to 'Add New Where Old Is Set'.

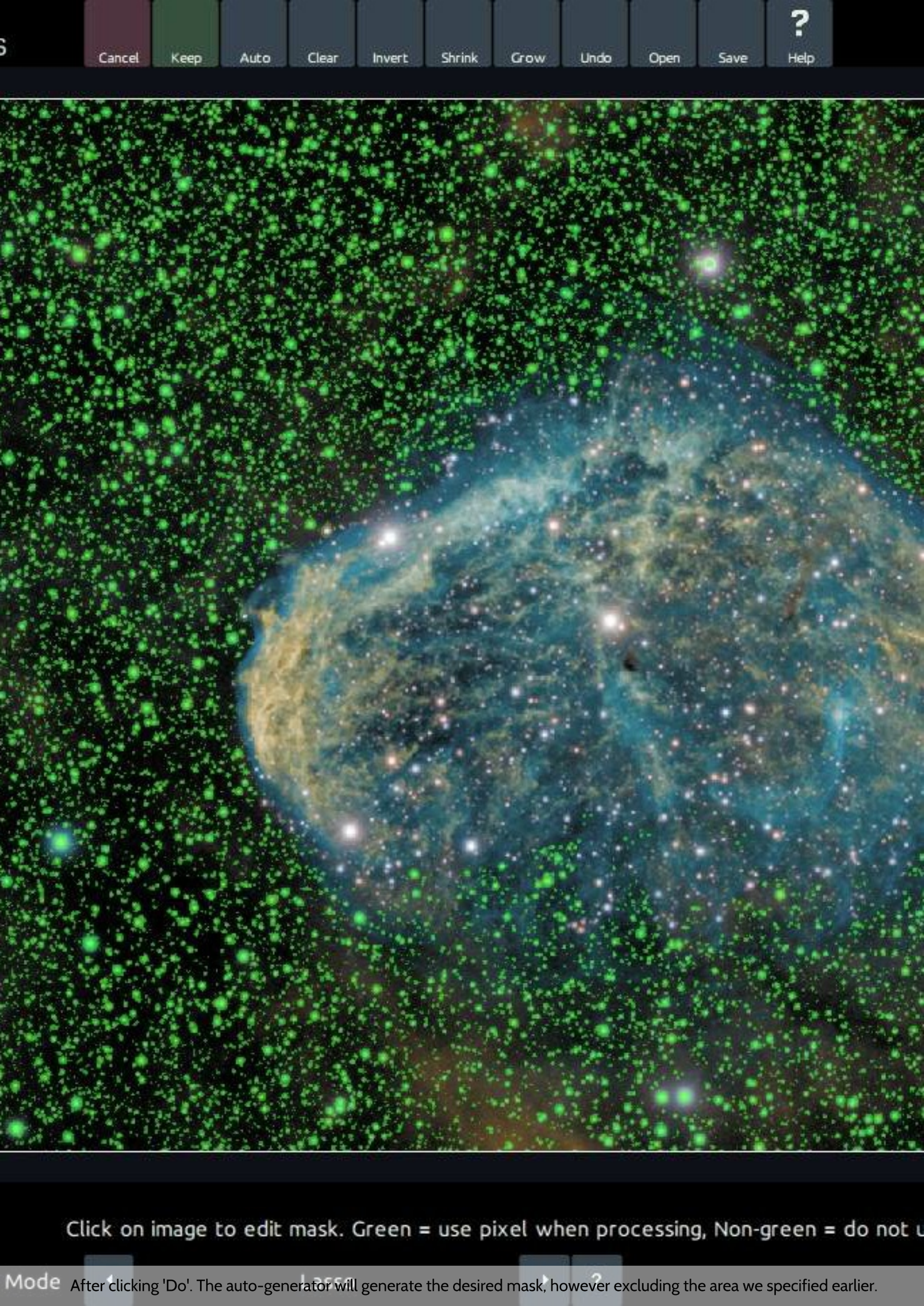
3 After clicking 'Do'. The auto-generator will generate the desired mask, however excluding the area we specified earlier.

4 Launch the Auto mask generator once more. Click the 'Stars' preset again. This time set 'Old Mask' to 'Add New To Old' to add the newly generated mask to the mask we already have. This will fill in the area we excluded earlier with the less aggressive mask as well.

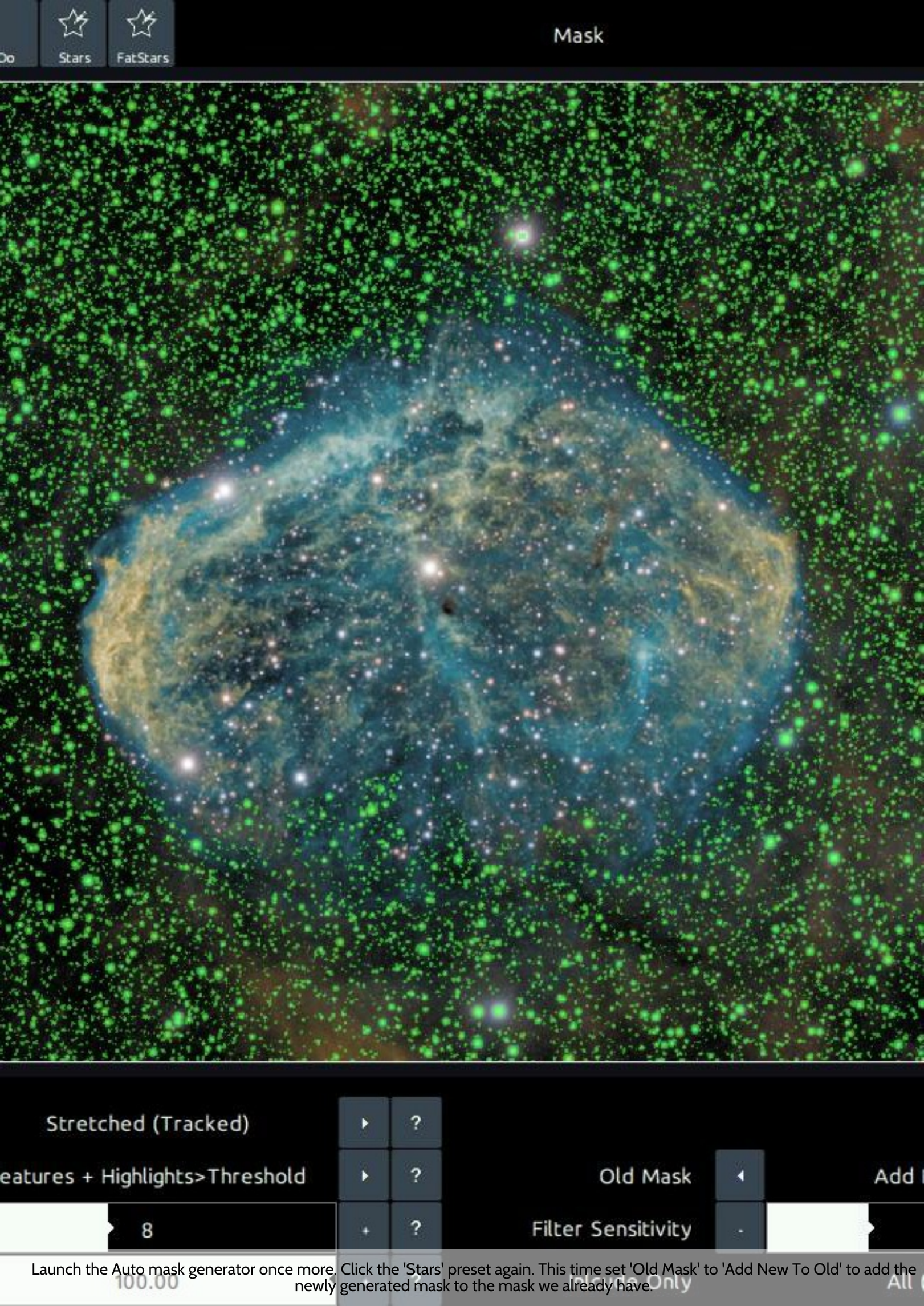



Clear the mask, and select the part of the image you wish to protect with the Flood Fill Lighter or Lasso tool, then click Invert.

In the Auto mask generator, set the parameters you need to generate your mask (here we choose the 'Stars' preset and set the 'Sourc
parameter to 'Stretched' to avoid any noise mitigation measures that may otherwise filter out faint stars for selection). Be sure to set
'Old Mask' to 'Add New Where Old Is Set'.




After clicking 'Do'. The auto-generator will generate the desired mask, however excluding the area we specified earlier.




Launch the Auto mask generator once more. Click the 'Stars' preset again. This time set 'Old Mask' to 'Add New To Old' to add the newly generated mask to the mask we already have.

We now have a mask that is less aggressive in the area we specified earlier, and more aggressive elsewhere.

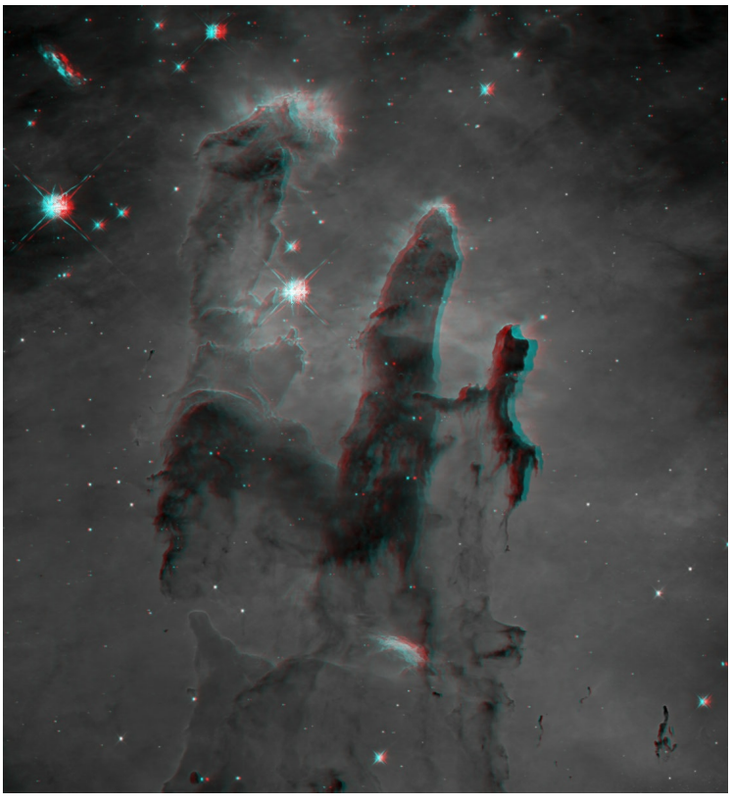
# Stereo 3D: Plausible depth information synthesis for 3D-capable media and Virtual Reality

New as of StarTools 1.6 beta, is the Stereo 3D module. The Stereo 3D module can be used to synthesise depth information based on astronomical image feature characteristics.

The depth cues introduced are merely educated guesses by the software and user, and should not be confused with scientific accuracy. Nevertheless, these cues can serve as a helpful tool for drawing attention to processes or features in an image.

Depth cues can also be highly instrumental in lending a fresh perspective to astronomical features in an image. The Stereo 3D module is able to generate plausible depth information for most deep space objects, with the exception of some galaxies.

The module can output various popular 3D formats, including side-by-side (for cross eye viewing), anaglyphs, depth maps, self-contained web content HTML, self-contained WebVR experiences and Facebook 3D photos.



The 3D Stereo module can output images in various formats, for example in this red/cyan anaglyph format.

# Usage

## Perceiving depth when using the module

Using the Stereo 3D module effectively starts with choosing a depth perception method that is most comfortable or convenient.

By default, the **Side-by-side Right/Left (Cross) Mode** is used, which allows for seeing 3D using the cross-viewing technique. If you are more comfortable with the parallel-viewing technique, you may select **Side-by-side Left/Right (Parallel)**. The benefits of the two aforementioned techniques is that they do not require any visual aids, while keeping coloring intact. The downside of these methods, is that the entire image must fit on half of the screen. E.g. zooming in breaks the 3D effect.

If you have a pair of red/cyan filter glasses, you may wish to use one of the three anaglyph **Modes**. The two monochromatic anaglyph modes render anaglyphs for printing and viewing on a screen. The screen-specific anaglyph will exhibit reduced cross-talk (aka "ghosting") in most cases. An "optimized" Color mode is also available, which retains some coloring. Visual spectrum astrophotography tends to contain few colors that are retained in this way, however narrowband composites can benefit. Finally, a **Depth Map** mode is available to inspect (or save) the z-axis depth information that was generated by the current model.

## Modelling and synthesizing depth information for astrophotography

The depth information generated by the Stereo 3D module is entirely synthetic and should not be ascribed any scientific accuracy. However, the modelling performed by the module is based on a number of assumptions that tend to hold true for many Deep Space Objects and can hence be used for making educated guesses about objects. Fundamentally, these assumptions are;

- Dark detail is visible by virtue of a brighter background. Dust clouds and Bok globules are good examples of matter obstructing other matter and hence being in the foreground of the matter they are obstructing.

- Brighter areas (for example due to emissions or reflection nebulosity) correlate well with voluminous areas.
- Bright objects within brighter areas tend to drive the (bright) emissions in their immediate neighborhoods. Therefore these objects should preferably be shown as embedded within these bright areas.
- Bright objects (such as bright blue O and B-class stars), drive emissions in their immediate neighborhood and tend to generate cavities due to radiation pressure.
- Stark edges such as shockfronts tend to speed away from their origin. Therefore these objects should perferably be shown as veering off.

## Tweaking the model

Depth information is created between two planes; the near plane (closest to the viewer) and the far plane (furthest away from the viewer). The distance between the two planes is governed by the '**Depth**' parameter.

The '**Protrude**' parameter governs the location of the near and far planes with respect to distance from the viewer. At 50% protrusion, half the scene will be going into the screen (or print), while the other half will appear to 'jut out' of the screen (or print). At 100% protrusion, the entire scene will appear to float in front of the screen (or print). At 0% protrusion the entire scene will appear to be inside the screen (or print).

The '**Luma to Volume**' parameter controls whether large bright or dark structures should be given volume. Objects that primarily stand out against a bright background (for example, the iconic Hubble 'Pillars of Creation' image) benefit from a shadow dominant setting. Conversely, objects that stand out against a dark background (for example M20) benefit from a highlight dominant setting.

The '**Simple L to Depth**' parameter naively maps a measure of brightness directly to depth information. This a somewhat crude tool and using the '**Luma to Volume**' parameter is often sufficient.

The '**Highlight Embedding**' parameter controls how much bright highlights should be embedded within larger structures and context. Bright objects such as energetic stars are often the cause of the visible emissions around them. Given they radiate in all directions, embedding them within these emission areas is the most logical course of action.

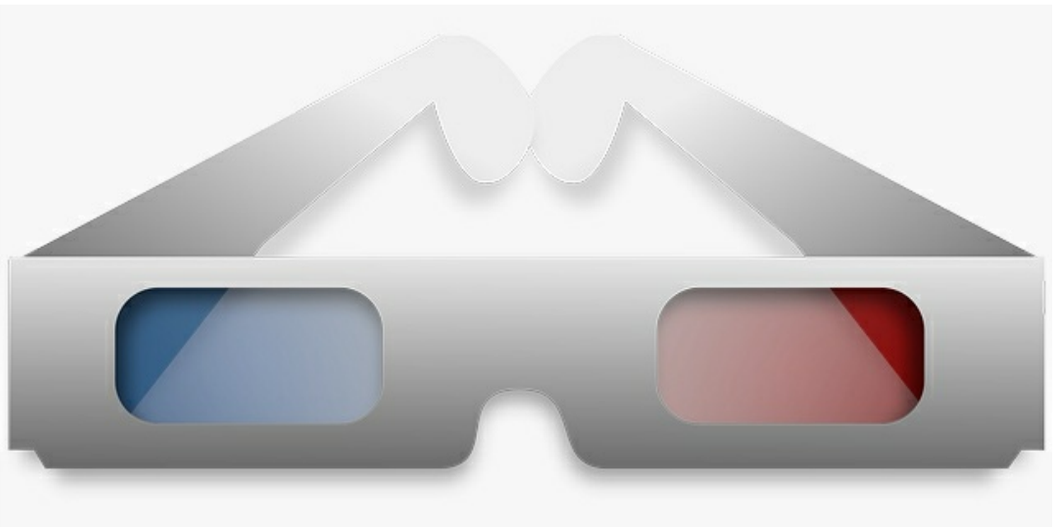
The '**Structure Embedding**' parameter controls how small-scale structures should behave in the presence of larger scale structures. At low values for this parameter, they tend to float in front of the larger scale structures. At higher values, smaller scale structures tend to intersect larger scale structures more often.

The '**Min. Structure Size**' parameter controls the smallest detail size the module may use to construct a model. Smaller values generate models suitable for widefields with small scale detail. Larger values may yield

more plausible results for narrowfields with many larger scale structures. Please note that larger values may cause the model to take longer to compute.

The '**Intricacy**' parameter controls how much smaller scale detail should prevail over larger scale detail. Higher values will yield models that show more fine, small scale changes in undulation and depth change. Lower values leave more of the depth changes to the larger scale structures.

The '**Depth Non-linearity**' parameter controls how matter is distributed across the depth field. Values higher than 1.0 progressively skew detail distribution towards the near plane. Values lower than 1.0 progressively skew detail distribution towards the far plane.



A pair of Red/Cyan glasses is a cheap but effective way of evaluating the output of the Stereo 3D module.

# Exporting to 3D-capable media

Besides rendering images as anaglyphs or side-by-side 3D stereo content, the Stereo 3D module is also able to generate Facebook 3D photos, as well as interactive self-contained 2.5D and Virtual Reality experiences.

## Standalone Virtual Reality experience

The 'WebVR' button in the module exports your image as a standalone HTML file. This file can be viewed locally in your webbrowser, or it can be hosted online.

It renders your image as an immersive VR experience, with a large screen wrapping around the viewer. The VR experience can be viewed in most popular headsets, including HTC Vive, Oculus, Windows Mixed Reality, GearVR, Google Day Dream and even sub-$5 Google Cardboard devices.

To view an experience, put it in an accessible location (locally or online) and launch it from a WebVR/XR capable browser.

Please note that landscape images tend to be more immersive.

The standalone VR experiences are compatible with anything from the latest headsets to the sub-$5 Google Cardboard devices.

Omega Nebula VR example

https://download.startools.org/swan_vr.html

## Standalone interactive 2.5D web viewer

The 'Web2.5D' button in the module exports your image as a standalone HTML file. This file can be viewed locally in your webbrowser, or it can be hosted online.

Depth is conveyed by a subtle, configurable, bobbing motion. This motion subtly changes the viewing angle to reveal more or less of the object, depending on the angle. The motion is configurable both by you and the viewer in both X and Y axes. The motion can also be configured to be mapped to mouse movements.

A so called 'depth pulse' can be sent into the image, which travels through the image from the near plane to the far plane, highlighting pixels of equal depth as it travels. The 'depth pulse' is useful to re-calibrate the viewer's persepective if background and foreground appear swapped.

Hosting the file online, allows for embedding the image as an IFRAME. The fol

```
<iframe scrolling="auto" marginheight="0" marginwidth="0" sty
spdx=4&amp;spdy=3&amp;caption=StarTools%20exports%20self-
contained,%20embeddable%20web%20content%20like%20this%20scene
frameborder="0"></iframe>
```

The following parameters can be set via the url;

- **modex**: 0=no movement, 1=positive sine wave modulation, 2=negative sine control
- **modey**: 0=no movement, 1=positive sine wave modulation, 2=negative sine
- **spdx**: speed of x-axis motion, range 1-5
- **spdy**: speed of y-axis motion, range 1-

5

- **caption**: caption for the image

Example of 2.5D embeddable web content

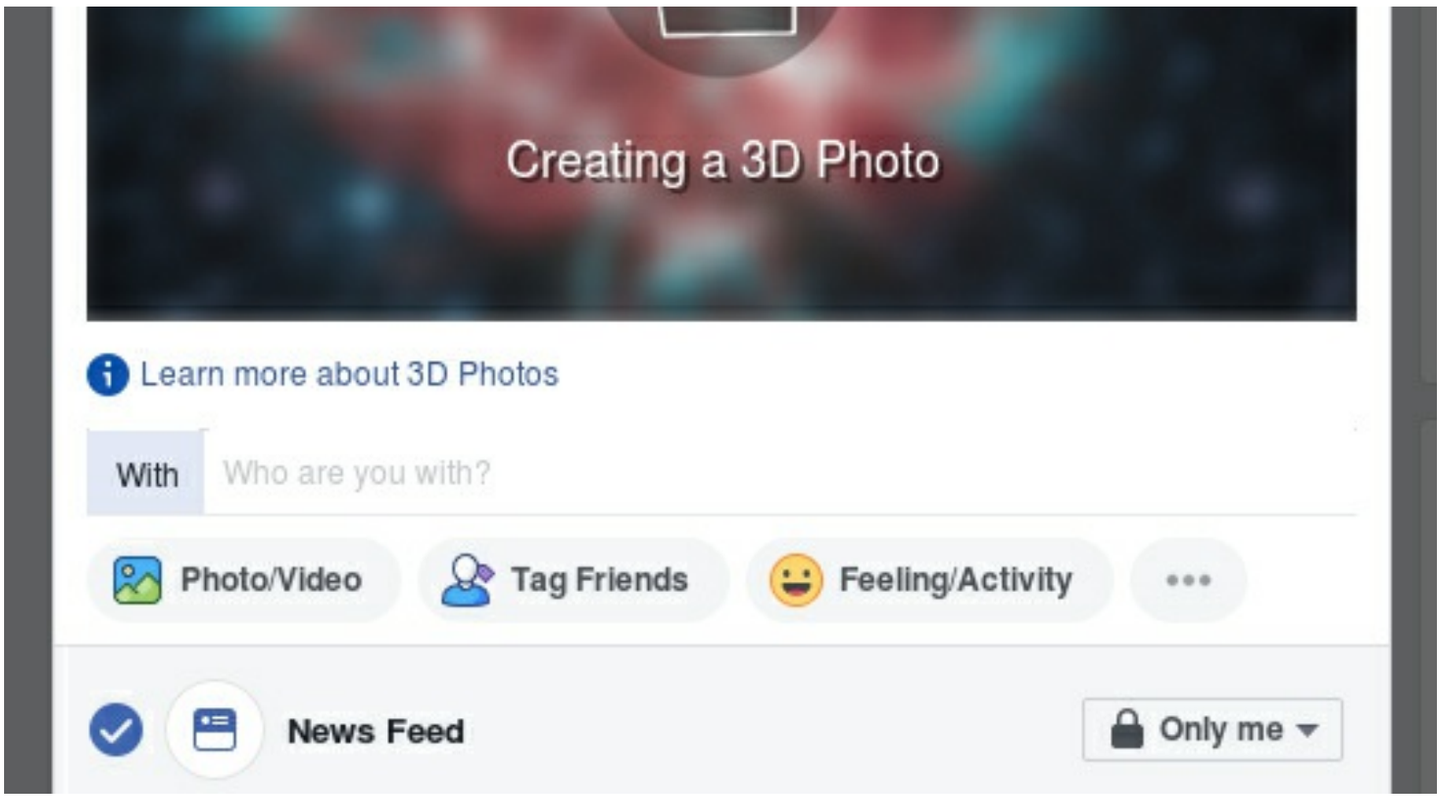
https://download.startools.org/pillars_stereo.html?spd
contained,%20embeddable%20web%20content%20l

## Facebook 3d photos

The Stereo 3D module is able to export your images for use with Facebook's 3

The 'Facebook' button in the module saves your image as dual JPEGs; one ima
*time* will see Facebook detect and use the two images to generate a 3D photo.

Please note that due Facebook's algorithm being designed for terrestrial photo
their halos. Nevertheless the result can look quite pleasing when simply brows




## 3D-capable TVs and projectors

TVs and projectors that are 3D-ready can - at minimum - usually be configured to render side-by-side images as 3D. Please consult your TV or projector's manual or in-built menu to access the correct settings.

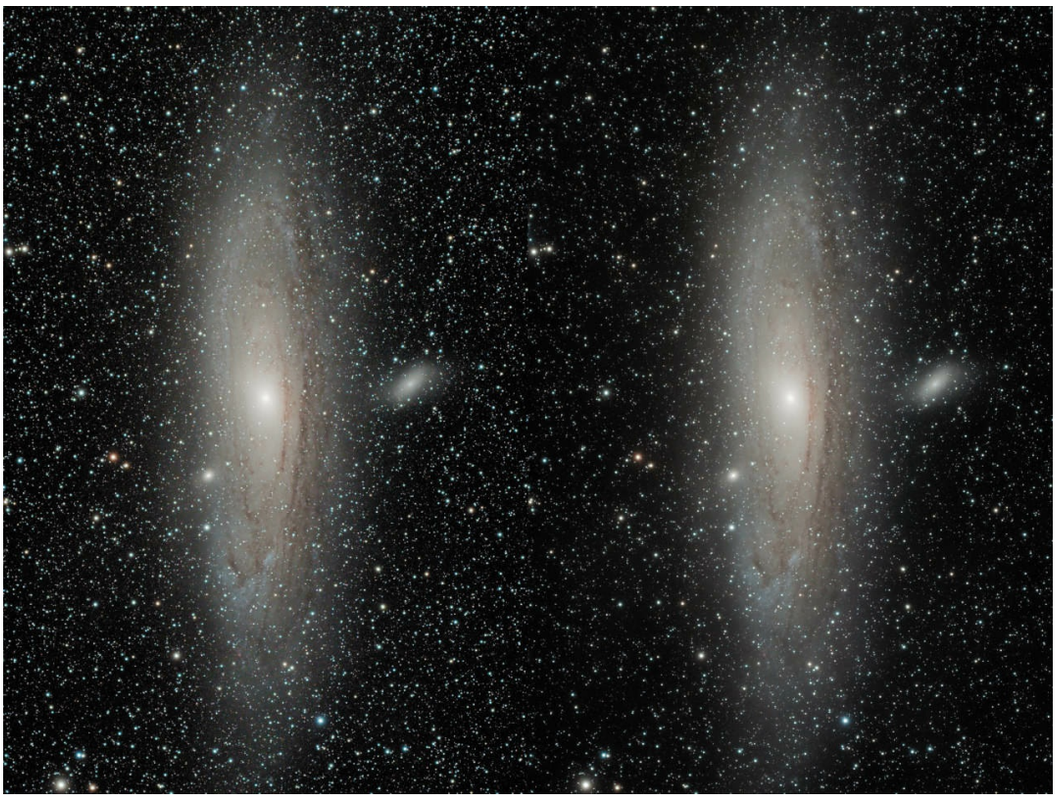
# Super Structure: Global Light Diffraction Remodelling of Large Scale Structures

The Super Structure allows you to manipulate the super structures in your image separately from the rest of the image. It is useful to push back busy star fields, or emphasise nebulosity by colour, luminance, or both.

The module brings back 'life' into an image by remodelling uniform light diffraction, helping larger scale structures such as nebulae and galaxies stand out and (re)take center stage; throughout the various processing stages, light diffraction (a subtle 'glow' of very bright objects due diffraction by a circular opening) tends to be distorted and suppressed through the various ways dynamic range is manipulated during processing. This can sometimes leave an image 'flat' and 'lifeless', or exaggerate the harshness of small stars.

The Super Structure module attempts to restore the effects of uniform light diffraction by an optical system, throughout a processed image, as if the image was recorded as-is. It does so by means of modelling an Airy disk pattern and re-calculating what the image would look like if it were diffracted by this pattern. The resulting model is then used to modulate or enhance the source image in various ways. The resulting output image

tends to have a re-established natural sense of depth and ambiance (as if looking at it through a telescope with the naked eye) with - if so desired - better visible super structures.



Left: input image. Right: DimSmall preset pushing back foreground stars and re-focusing the attention on M31.

# Usage

As with most modules in StarTools, the Super Structure module comes with a number of presets;

- 'DimSmall' pushes back anything that is not a super structure while retaining energy allocated to super structures. Overall image brightness is compensated for.

- 'Brighten' brightens detected super structures.
- 'Isolate' is similar to the 'DimSmall' preset, however does not compensate for lost energy (image brightness).
- 'Airy Only' shows the AiryDisk model only for fine tuning or use in other ways.
- 'Saturate' saturates the colours of detected super structures.

The '**Strength**' parameter governs the overall strength of the effect.

The '**Brightness, Color**' parameter determines whether brightness, colour or both is affected.

The '**Saturation**' parameter controls the colour saturation of the output model (viewable by using the 'AiryOnly' preset), before it is composited with the source image to generate the final output.

The '**Detail Preservation**' parameter selects the detail preservation algorithm the Super Structure module should use to merge the model with the source image to produce the output image;

- 'Off' does not attempt to preserve any detail.
- 'Min Distance to 1/2 Unity' uses whichever pixel that is closest to half unity (e.g. perfect gray).

- 'Linear Brightness Mask' uses a brightness mask that progressively masks-out brighter values until it uses the original values instead.
- 'Linear Brightness Mask Darken' uses a brightness mask that progressively masks out brighter values. Only pixels that are darker than the original image are kept.

The '**Detail Preservation Radius**' sets a filter radius that is used for smoothly blending processed and non-processed pixels, if the '**Detail Preservation**' parameter is set to 'Min Distance to 1/2 Unity'. It is grayed out otherwise.

The '**Compositing Algorithm**' parameter defines how the calculated diffraction model is to be generally combined with the original image:

- 'None (Output Super Structure Only)' outputs the Super Structure model only and does not composite it with the source image.
- 'Screen' works like projecting two images on the same screen; the input image and the Super Structure model.
- 'Power of Inverse' composites the original image with the Super Structure model using a Power of Inversed Pixels (PIP) function.
- 'Multiply, Gamma Correct' multiplies the original image with the Super Structure model and then takes the square root.
- Multiply, 2x Gamma Correct - similar to 'Multiply, Gamma Correct' but

doubles the Gamma
Correction.

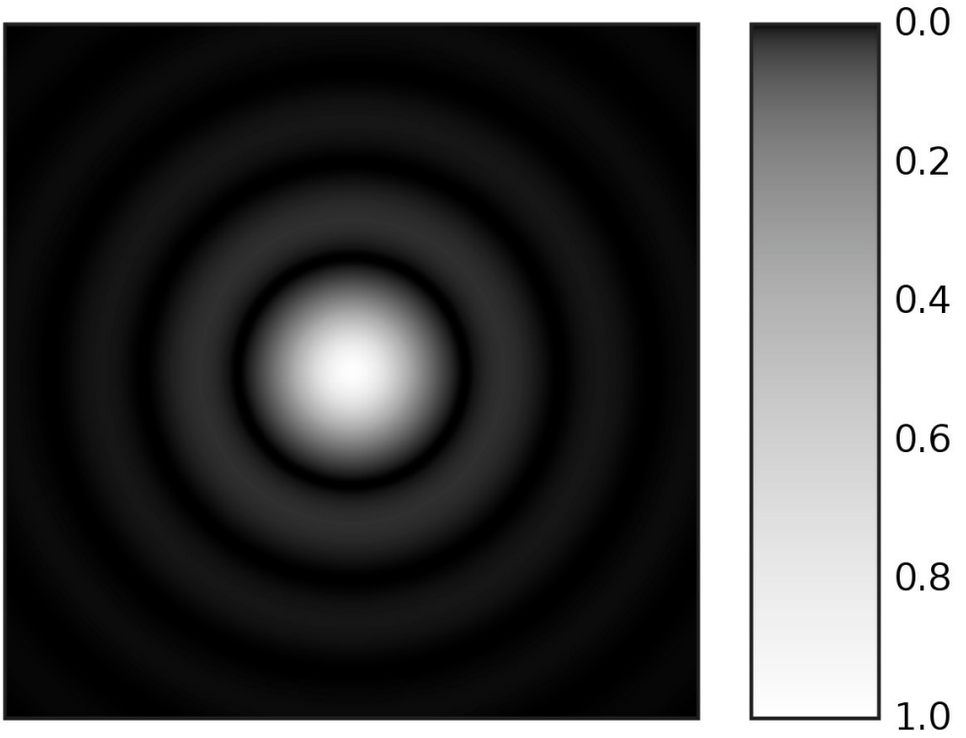
The '**Airy Disk Radius**' parameter sets the radius of the Airy disk point spread function (PSF) that is used to diffract the light. Smaller values are generally more suited to wide fields, whereas larger values are generally best for narrow fields. This is so that the PSF mimics the diffraction pattern of the original optical train. 'Incorrect' values may make the image look fuzzier than need be (in the case of wide fields), or may define super structures less well (in the case of narrow fields).

The '**Brightness Retention**' feature attempts to retain the apparent brightness of the input image. In the case of 'Local Median', a local median value is calculated for each pixel that is used as the target brightness value to which the modifications are added. In the case of 'Global Mode Align, Darken Only' it retain brightness by calculating a non-linear stretch that aligns the histogram peak (statistical 'mode') of the old image with that of the new image. After doing so, a 'Darken Only' operation only keeps pixels from the resulting image that are darker than the input image.

Finally, as with most modules in StarTools that employ masks, a '**Mask Fuzz**' parameter is available to smoothly blend the transition between masked and non-masked pixels. Note that the Super Structure module may - as a last resort - be used locally by means of a mask. In this case

the Super Structure module can be used to isolate objects in an image and lift them from an otherwise noisy background.

By having the Super Structure module augment an object's super-structure, faint objects that were otherwise unsalvageable can be made to stand out from the background. Please note that, depending on the nature of the used selective mask, the super structures introduced by using the Super Structure module in this particular way with a selective mask, should be regarded as an educated guess rather than documentary detail, and technically falls outside of the realm of documentary photography.




A computer-generated image of an Airy disk. The grayscale intensities have been adjusted to enhance the brightness of the outer rings of the Airy pattern. Source: Wikipedia.

3 examples of the Super Struture module presets manipulating the visibility of large scale structures. Top left; original. Top right; 'Brighten' preset. Bottom left; 'DimSmall' preset. Bottom right; 'Saturate' preset.

# SVDecon: Detail Recovery through Spatially Variant Distortion Correction

StarTools is the first and only software for astrophotography to implements true, fully generalised Spatially Variant PSF deconvolution (aka "anisotropic" or "adaptive kernel" deconvolution). The fully GPU accelerated solution is robust in the face of even severe noise, meaning it can deployed to restore detail in almost real-time in almost every dataset.

Even the best optical systems will suffer from minute differences in Point Spread Functions (aka "blur functions") across the image. Therefore, a generalised deconvolution solution that can take these changing distortions into account, has been one of the holy grails of astronomical image processing.

## Innovations at a glance

The SVDecon module incorporates a series of unique innovations that sets it apart from all other legacy implementations as found in other software;

- It corrects for *multiple, different* distortions at

different locations in the dataset, rather than just *one* distortion for the entire dataset

- It ***preferably*** operates on highly processed and stretched data (provided StarTools' signal evolution Tracking is engaged)
- It performs intra-iteration resampling of PSFs
- It is almost always able to provide meaningful improvements, even when dealing with marginal datasets and signals
- It is robust in the presence of severe noise, as well as natural singularities (e.g. over-exposed star cores) in the dataset
- Depending on your system, previews complete in near-real-time
- Any development of noise grain is tracked and marked for removal/mitigation during final noise reduction
- Smart caching allows faster tweaking of some parameters (such as de-ringing) without needing re-doing full deconvolution
- Doing all this, the algorithm at its core, is still based on true Richardson & Lucy deconvolution, and thus its behavior is well understood, documented and accepted in the scientific community, as opposed to black-box neural hallucination-based image re-interpretation algorithms.

Unique amongst its peers, the GPU-accelerated SVDecon module in StarTools, is robust in the face of severe noise, singularities (such as over-exposing star cores), extreme non-linear processing and local detail enhancement, and variable Point Spread Functions. Left: original at 200% zoom. Right: deconvolved with Spatially Variant PSF Deconvolution at 200% zoom.

# Usage

It is important to understand two things about deconvolution as a basic, fundamental process;

- Deconvolution is "an ill-posed problem", due to the presence of noise in every dataset. This means that there is no one perfect solution, but rather a range of approximations to the "perfect" solution.
- Deconvolution should *not* be confused or equated with sharpening; deconvolution should be seen as a means to *restore* a compromised (distorted by atmospheric turbulence and/or diffraction by the optics)

dataset. It is not meant as an acuity enhancing process or some sort of beautification filter. You should (will) always be able to corroborate the detail it restores, using the work from your peers, observatories and space agencies.

In addition to the above, deconvolution with a *spatially variant* Point Spread Function, adds to the complexity of basic deconvolution by requiring a model that accurately describes how the Point Spread Function changes *across* the image, rather than assuming a one-distortion-fits-all.

Understanding these important points will make clear why some of the various parameters exist in this module, and what is being achieved by the module.

Even this heavily distorted dataset with changing trailing direction from corner-to-corner can be improved.

## Modes of operation

The SVDecon module can operate in several implicit modes, depending on how many star samples - if any - are provided;

- When no star samples are provided, the SVDecon module will operate in a similar way to the pre-1.7 deconvolution modules.; a selection of synthetic models are available to model *one* specific atmospheric or optical distortion that is true for the *entire* image.
- When one star sample is provided, the SVDecon module will operate in a way similar to the 1.7 module (though somewhat more effectively); a *single* sample provides the atmospheric distortion model for the entire image, while an optional synthetic optics model provides further refinement.
- When multiple star samples are provided, the SVDecon module will operate in the most advanced way. Multiple samples provide a distortion model that *varies per location* in the image. An optional optical synthetic model may be used for further refinement, though is usually best turned off.

The latter mode of operation is usually the preferred and recommended way of using the module, and takes full advantage of the module's unique spatially variant PSF modelling and correction capabilities.

The module automatically grays out parameters that are not being used,

and may also change (zero-out or disable) some parameters in line with the different modes as they are accessed.

When the subject is lunar or planetary in nature, no star samples are typically available. The "**Planetary/Lunar**" preset button configures the module for optimal use in these situations.

Finally, details of the mode being used, are reflected in the message below the image window.

## Apodization Mask

The SVDecon module requires a mask that marks the boundaries of stellar profiles. Pixels that fall inside the masked areas (designated "green" in the mask editor), are used during local PSF model construction. Pixels that fall outside the masked area are disregarded during local PSF model construction.

It is highly recommended to to include as much of a star's stellar profile in the mask as possible. Failure to do so may lead to increased ringing artifacts around deconvolved stars. Sometimes a simple manual "Grow" operation in the mask editor suffices, in order to include more of the stellar profiles.

Compared to most other deconvolution implementations, the SVDecon module is robust in the face of singularities (for example over-exposing star cores). In fact, it is able to coalesce such singularities further. As such,

the mask is no longer primarily used for designating singularities in the image, like it was in versions of StarTools before version 1.8.

The mask does however double as a rough guide for the de-ringing algorithm, indicating areas where of ringing may develop. Clearing the mask (all pixels off/not green in the mask editor) is generally recommended for non-stellar objects, including lunar, planetary or solar data. As a courtesy, this clearing is performed automatically when selecting the **Planetary/Lunar** preset.

## Point Spread Functions (PSFs)

A deconvolution algorithm's task, is to reverse the blur caused by the atmosphere and optics. Stars, for example, are so far away that they should really render as single-pixel point lights. However in most images, stellar profiles of non-overexposing stars show the point light spread out across neighbouring pixels, yielding a brighter core surrounded by light tapering off. Further diffraction may be caused by spider vanes and/or other obstructions in the Optical Tube Array, for example yielding diffraction spikes. Even the mere act of imaging through a circular opening (which is obviously unavoidable) causes diffraction and thus "blurring" of the incoming light.

The point light's energy is scattered/spread around its actual location, yielding the blur. The way a point light is blurred like this, is also called a Point Spread Function (PSF). Of course, *all* light in your image is spread according to a Point Spread Function (PSF), not just the stars.

Deconvolution is all about modelling this PSF, then finding and applying its reverse to the best of our abilities.

Introducing Spatial Variance

Traditional deconvolution, as found in all other applications, assumes the Point Spread Function is the same across the image, in order to reduce computational and analytical complexity. However, in real-world applications the Point Spread Function will vary for each (X, Y) location in a dataset. These differences may be large or small, however always noticeable and present; no real-world optical system is perfect. Ergo, in a real-world scenario, a Point Spread Function that perfectly describes the distortion in one area of the dataset, is typically incorrect for another area of that same dataset.

Traditionally, the "solution" to this problem has been to find a single, best-compromise PSF that works "well enough" for the entire image. This is necessarily coupled with reducing the amount of deconvolution possible before artifacts start to appear (due to the PSF not being accurate for all areas in the dataset).

Being able to use a *unique* PSF for every (X, Y) location in the image solves aforementioned problems, allowing for superior recovery of detail without being limited by artifacts as quickly.

### Synthetic vs sampled PSFs

The SVDecon module, makes a distinction between two types of Point Spread Functions; synthetic and sampled Point Spread Functions. Depending on the implicit mode the module operates in, synthetic, sampled, or both synthetic *and* sampled PSFs are used.

When no samples are provided (for example on first launch of the SVDecon module), the module will fall back on a purely synthetic model for the PSF. As mentioned before, this mode uses the single PSF for the entire image. As such the module is not operating in its spatially variant mode, but rather behaves like a traditional, single-PSF model, deconvolution algorithm as found in all other software. Even in this mode, its results should be superior to most other implementations, thanks to signal evolution Tracking directing artefact suppression.

A number of parameters can be controlled separately for the synthetic and sampled Point Spread Function deconvolution stages.

## Synthetic PSFs

### Synthetic PSF models

Atmospheric and lens-related blur is easily modelled, as its behaviour and effects on long exposure photography has been well studied over the decades. 5 subtly different models are available for selection via the '**Synthetic PSF Model**' parameter;

- 'Gaussian' uses a Gaussian distribution to model atmospheric blurring.
- 'Circle of Confusion' models the way light rays from a lens are unable to come to a perfect focus when imaging a point source (aka the 'Circle of Confusion'). This distribution is suitable for images taken outside of Earth's atmosphere or images where Earth's atmosphere did otherwise not distort the image.
- 'Moffat Beta=4.765 (Trujillo)' uses a Moffat distribution with a Beta factor of 4.765. Trujillo et al (2001) propose in their paper that this value (and its resulting PSF) is the best fit for prevailing atmospheric turbulence theory.
- 'Moffat Beta=3.0 (Saglia, FALT)' uses Moffat distribution with a Beta factor of 3.0, which is a rough average of the values tested by Saglia et al (1993). The value of ~3.0 also corresponds with the findings Bendinelli et al (1988) and was implemented as the default in the FALT software at ESO, as a result of studying the Mayall II cluster.
- 'Moffat Beta=2.5 (IRAF)' uses a Moffat distribution with a Beta factor of 2.5, as implemented in the IRAF software suite by the United States National Optical Astronomy Observatory.

Only the 'Circle of Confusion' model is available for further refinement when samples are available. This allows the user to further refine the sample-corrected dataset if desired, assuming any remaining error is the result of 'Circle of Confusion' issues (optics-related) with all other issues

corrected for as much as possible.

The PSF radius input for the chosen synthetic model, is controlled by the '**Synthetic PSF Radius**' parameter. This parameter corresponds to the approximate the area over which the light was spread; reversing a larger 'blur' (for example in a narrow field dataset) will require a larger radius than a smaller 'blur' (for example in a wide field dataset).

The '**Synthetic Iterations**' parameter specifies the amount of iterations the deconvolution algorithm will go through, reversing the type of synthetic 'blur' specified by the '**Synthetic PSF Model**'. Increasing this parameter will make the effect more pronounced, yielding better results up until a point where noise gradually starts to increase. Find the best trade-off in terms of noise increase (if any) and recovered detail, bearing in mind that StarTools signal evolution Tracking will meticulously track noise propagation and can snuff out a large portion of it during the Denoise stage when you switch Tracking off. A higher number of iterations will make rendering times take longer - you may wish to use a smaller preview in this case.

## Sampled PSFs

Sampled PSF models

Ideally, rather than relying on a *single* synthetic PSF, multiple Point Spread Functions are provided instead, by means of carefully selected samples. These samples should take the form of. isolated stars on an even background that do not over expose, nor are too dim. Ideally, these samples are provided for all areas across the image, so that the module can analyse and model how the PSF changes from pixel-to-pixel for all areas of the image.

## Recommended workflow

As opposed to all other implementations of deconvolution in other software, the usage of the SVDecon module is generally recommended towards *the end* of your luminance (detail enhancement) processing workflow. That is, ideally, you will have already carried out the bulk of your stretching and detail enhancement before launching the SVDecon module. The reason for this, is that the SVDecon module makes extensive use of knowledge that indicates *how* you processed your data prior to invoking it, and how detail evolved and changed during your processing. This knowledge specifically feeds into the way noise and artifacts are detected and suppressed during the regularisation stage for each iteration.

For most datasets, superior results are achieved by using the module in

Spatially Variant mode, e.g by providing multiple star samples. In cases where providing star samples is too difficult or time consuming, the default synthetic model will still very good results however.

Selecting samples for Spatially Variant deconvolution

To provide the module with PSF samples, the '**Sampling**' view should be selected. This view is accessed by clicking the '**Sampling**' button in the top right corner. This special was designed to help the user identify and select good quality star samples.

In the '**Sampling**' view, A convenient rendering of the image is shown, in which;

- Candidate stars are delineated by an outline.
- Red pixels show low quality areas
- Yellow pixels show borderline usable areas.
- Green pixels show high quality areas.

Ideally, you should endeavour to find stars samples that have a green inner core without any red pixels at their centre. If you cannot find such stars and you need samples in a specific area you may choose samples that have a yellow core instead. As a rule of thumb, providing samples in all areas of the image takes precedence over the quality of the samples.

You should avoid;

- Stars that sit on top of nebulosity or other detail.
- Objects that are not stars (for example distant galaxies)
- Stars that are close to other stars
- Stars that appear markedly different in shape compared to other stars nearby
- Stars whose outline appear non-oval or concave or markedly different to the outlines of other stars nearby

Star samples can be made visible on the regular view (e.g. the view with the before/after deconvolved result) by holding the left mouse button. Star samples will also be visible outside any preview area, this also doubles as a reminder that any selected PSF Resampling algorithm will not resample those stars (see 'PSF resampling mode'). You may also quickly de-select stars via the regular before/after view by clicking on a star that has a sample over it that you wish to remove.

## The Sampled Area

The immediate area of a sampled star is indicated by a blue square ('bounding box'). This area is the '**Sampled Area**'. A sampled area should contain *one* star sample only; you should avoid selecting samples that have parts of other stars in the blue square surrounding a prospective sample. The size of the blue square is determined by the '**Sampled Area**' parameter. The '**Sampled Area**' parameter should be set in such a way that all samples' green pixels fall well within the blue area's confines and are not 'cut-off' by the blue square's boundaries.

## Star sample outlines and apodization mask

The star sample outlines are constructed using the apodization mask that is generated. You may touch up this mask to avoid low-quality stars being included in the blue square '**Sampled Area**', if that helps to better sample a high quality star.

## Number of samples and location of samples

Ideally samples are specified in all areas of the image in equal numbers. The module will work with any amount of samples, however ten or more, good quality samples is recommended. The amount of samples you should provide is largely dependent on how severe the distortions are in the image and how they vary across the image.

Please note that, when clicking a sample, the indicated centre of a sample will not necessarily be the pixel you clicked, nor necessarily the

brightest pixel. Instead, the indicated centre is the ''luminance centroid''. It is the weighted (by brightness) mean of all pixels in the sample. This is so that, for example, samples of stars that are deformed or heavily defocused (where their centre is less bright than their surroundings) are still captured correctly.

Heavily distorted PSFs

For images with heavily distorted PSFs that are highly variant (for example due to field rotation, tracking error, field curvature, coma, camera mounting issue, or some other acquisition issue that has severely deformed stars in an anisotropic way), the **'Spatial Error'** parameter may need to be increased, with the **'Sampled Iterations'** increased in tandem. The **'Spatial Error'** parameter relaxes locality constraints on the recovered detail, and increasing this parameter, allows the algorithm to reconstruct point lights from pixels that are much less co-located than would normally be the case. Deconvolution is not a 100% cure for such issues, and its corrective effect is limited by what the data can bear without artifacts (due to noise) becoming a limiting factor.

Under such challenging conditions, improvement should be regarded in the context of improved detail, rather than perfectly point or circle-like stellar profiles. While stars may definitely become more pin-point and/or 'rounder', particularly areas that are (or are close to) over-exposing, such as very bright stars, may not contain enough data for reconstruction due to clipping or non-linearity issues. Binning the

resulting image slightly afterwards, may somewhat help hide issues in the stellar profiles. Alternatively, the Repair module may help correcting these stars.

PSF Resampling mode

The SVDecon module is innovative in many ways, and one such innovation is its ability to re-sample the stars *as they are being deconvolved*. This feedback tends to reduce the development of ringing artifacts and can improve results further.

Three '**PSF Resampling**' modes are available;

- None; no resampling and model reconstruction occurs during deconvolution - the samples are used as-is.
- Intra-Iteration; all samples are resampled at their original locations for each iteration
- Intra-Iteration + Centroid Tracking; all samples are resampled after their locations have first been re-determined.Intra-iteration resampling while a preview is being used will only re-sample the samples that are contained within the preview. Therefore, the full effects of intra-iteration resampling are best evaluated without a preview being defined. As, depending on your system's CPU and GPU resources, intra-iteration resampling may be rather taxing, it may be useful to evaluate its effects only once all samples are set and once you are happy with the results without PSF resampling activated.

## Dynamic Range Extension

The '**Dynamic Range Extension**' parameter provides any reconstructed highlights with 'room' to show their detail, rather than clipping themt against the white point of the input image. Use this parameter if significant latent detail is recovered that requires more dynamic range to be fully appreciated. Lunar datasets can often benefit from an extended dynamic range allocation.

## Planetary, solar and lunar datasets

A preset for lunar, planetary, solar use quickly configures the module for lunar, planetary and solar purposes; it clears the apodization mask (no star sampling possible/needed) and dials in a much higher amount of iterations. It also dials in a large synthetic PSF radius more suitable to reverse atmospheric turbulence-induced blur for high magnification datasets. You will likely want to increase the amount of iterations further, as well as adjust the PSF radius to better model the specific seeing conditions.

## Evaluating the result

A considerable amount of research and development has gone into CPU and GPU optimisation of the algorithm; an important part of image processing is getting accurate feedback as soon as possible on decisions made, samples set, and parameters tweaked.

As a result, it is possible to evaluate the result of including and excluding samples in near-real-time; you do not need to wait minutes for the algorithm to complete. This is particularly the case when a smaller preview area is selected.

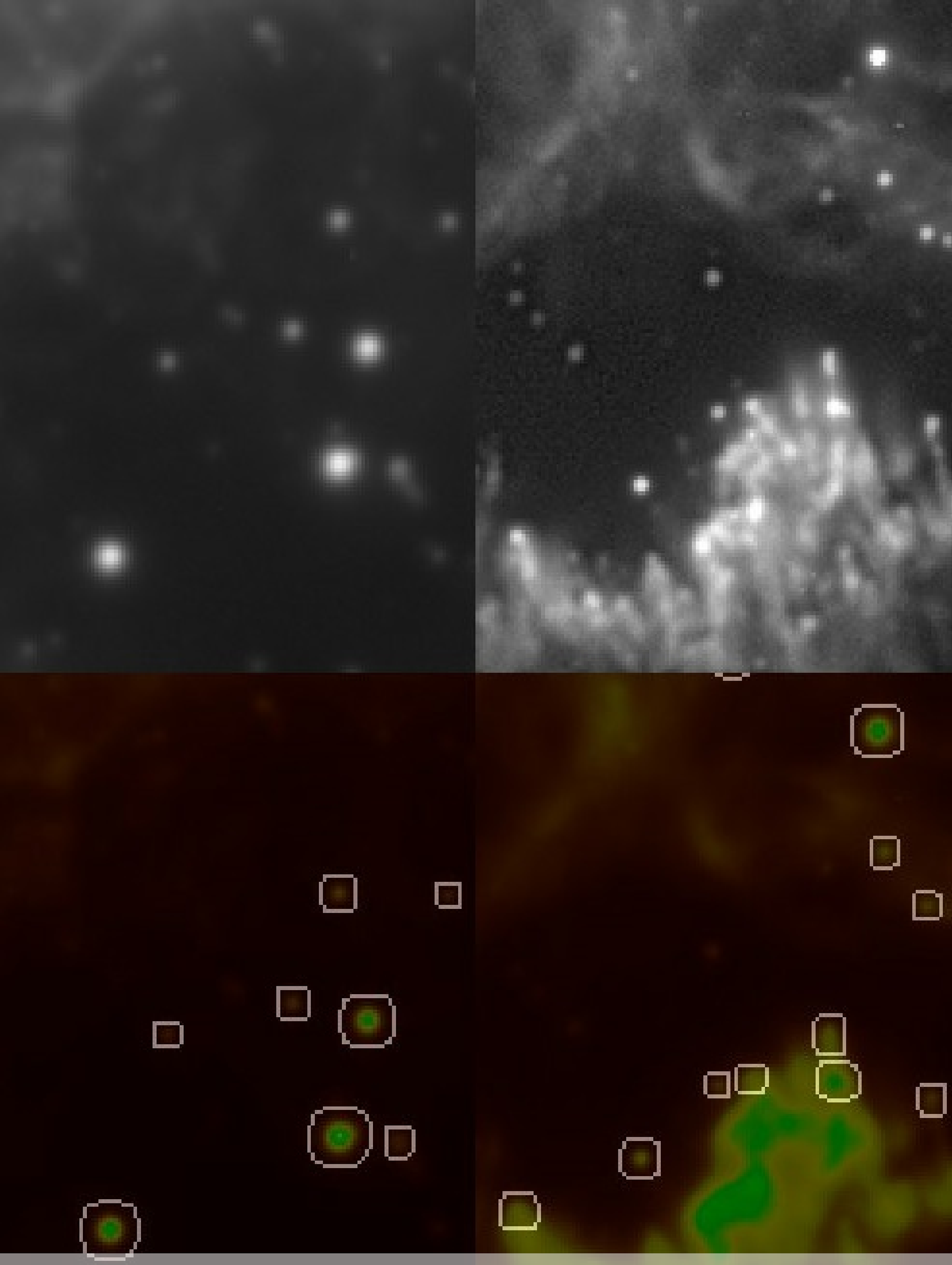
As stated previously, please note however, that the '**PSF Resampling**' feature is only carried out on any samples that exist in the preview area. As a result, when a '**PSF Resampling**' mode is selected, previews may differ somewhat from the full image. To achieve a preview for an area when a '**PSF Resampling**' mode is selected, try to include as many samples in the preview area as possible when defining the preview area's bounding box.

With the aforementioned caveat with regards to resampling in mind however, any samples that fall outside the preview are still used for construction of the local PSF models for pixels inside the preview. In other words, the results in the preview should be near-identical to deconvolution of the full image, unless a specific '**PSF Resampling**' mode is used.

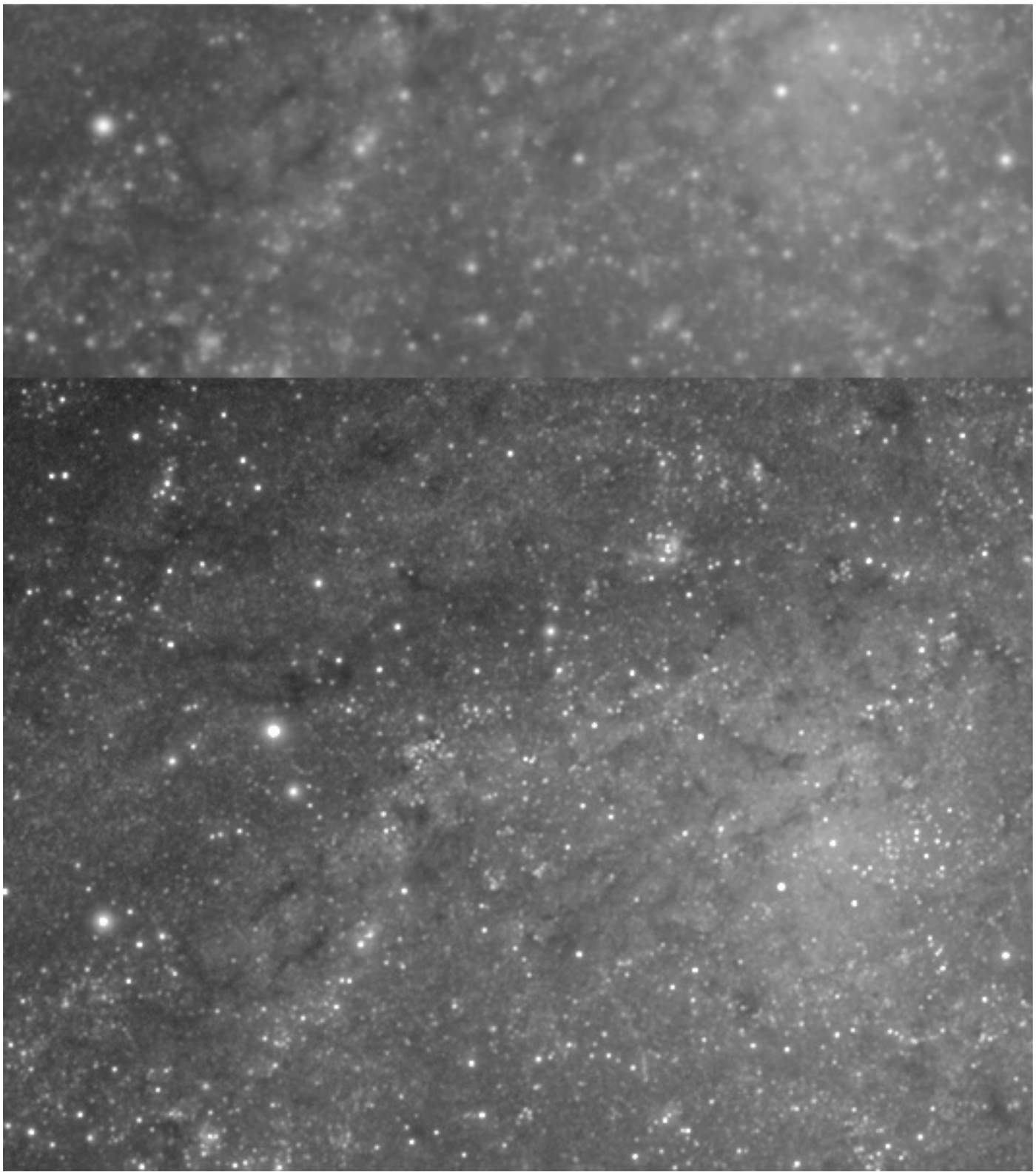
Noise and artifact propagation

While it is best to avoid overly aggressive settings that exacerbate noise grain (for example by specifying a too large number of iterations), a significant portion of such grain will be still be very effectively addressed during the final noise reduction stage; StarTools' Tracking engine will have pin-pointed the noise grain and its severity and should be able to significantly reduce its prevalence during final noise reduction (e.g. when switching Tracking off).

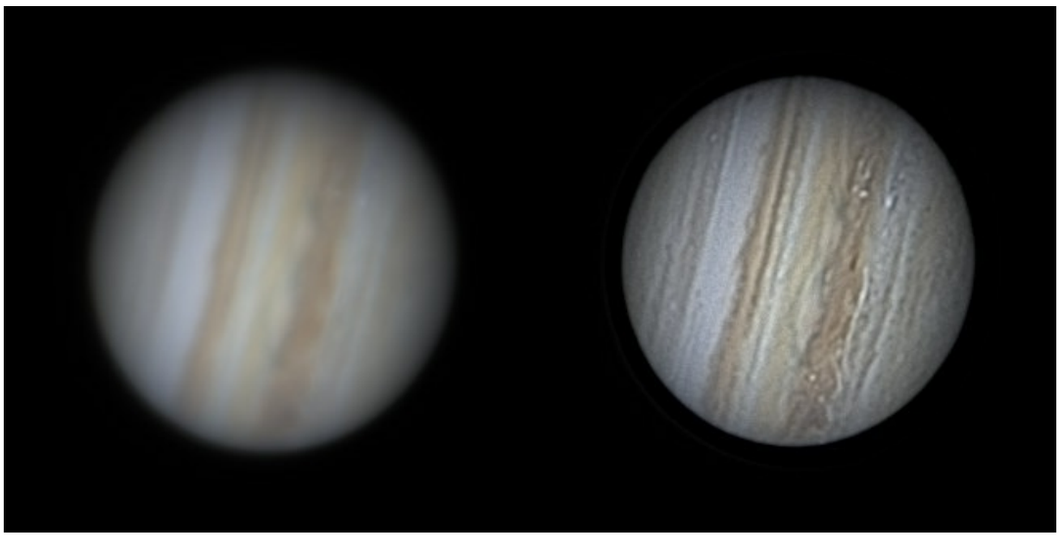
Ringing artifacts and/or singularity-related artifacts are harder to address and their development are best avoided in the first place by choosing appropriate settings. As a last resort, the '**Deringing Amount**', '**Deringing Detect**' and '**Deringing Fuzz**' parameters can be used to help mitigate their prevalence.



The SVDecon module is ideally operated by selecting samples of good quality. Top left; 200% zoom original image. Top right; the resulting deconvolved image by selecting the samples indicated. Bottom left; "Sample Quality" view without samples selected. Bottom right; "Sample Quality" view with three good quality samples selected. The blue bounding boxes should ideally fit the entire green "blobs" (signifiying the apodization mask for each sample).



Detail should snap into focus, stars should coalesce into point lights and halos around over-exposing stars should be diminished.



Just using synthetic PSF modelling, the SVDecon module is just as effective on lunar, planetary and solar datasets, as it is on deep space datasets.

SVDecon can greatly improve the clarity of image data from even space telescopes and probes such as the Hubble Space Telescope
Left column: original data, Center column: SV deconvolved data, Right column: default noise mitigation during final signal Tracking
Denoise stage, demonstrating precise, autonomous deconvolution-induced grain and artifact tracking.

## Recovering PSF samples from the log

Any samples you set, are stored in the StarTools.log file and can be restored using the '**LoadPSFs**' button.

In the StarTools.log file, you should find entries like these;

```
PSF samples used (8 PSF sample locations, BASE64
encoded)
VFMAAAgAOAQMA/oDEQHaAoEAIwNeAOQAUwDUAY8AbAI5AdMBMQGkAFAB
```

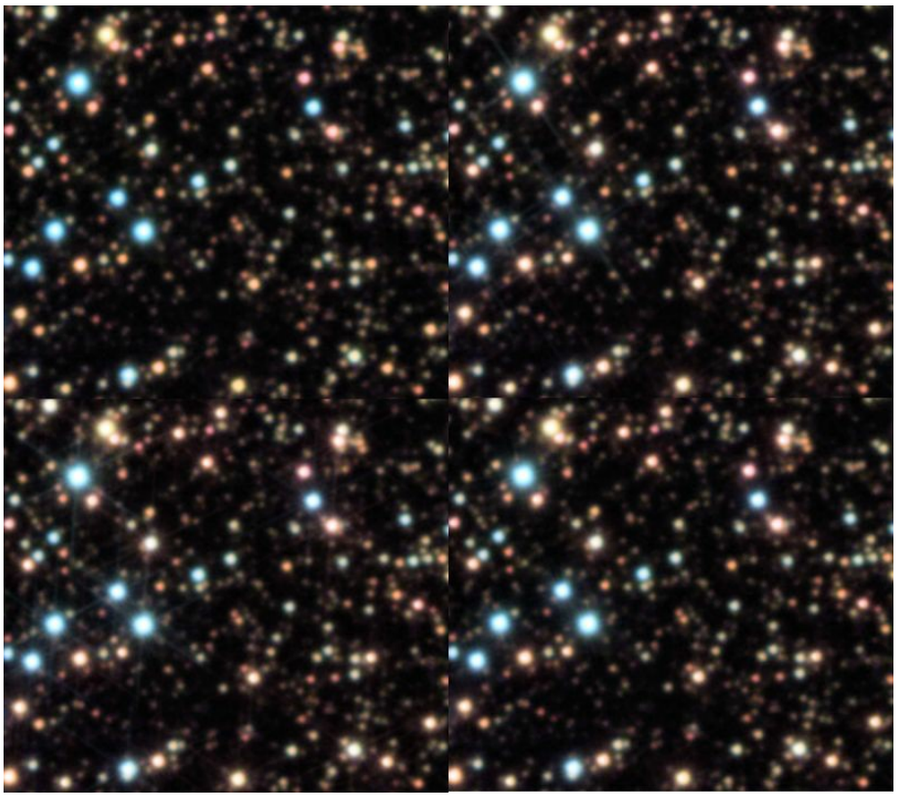
If you wish to restore the samples used, put the BASE64 string (starting with VFM... in the example) in a text file. Simply load the the file using the '**LoadPSFs**' button.

# Synth: Star Resynthesis and Augmentation

The Synth module generates physically correct diffraction and diffusion of point lights (such as stars) in your image, based on a virtual telescope model.

Besides correcting and enhancing the appearance of point lights (such as stars), the Synth module may even be 'abused' for aesthetic purposes to endow stars with diffraction spikes where they originally had none. It is worth noting that any other tools on the market today simply approximate the visual likeness of such star spikes and 'paint' them on. However the Synth module can physically model and emulate most real optical systems and configurations to obtain a desired result.

While synthetic PSF augmentation has since been used on Hubble data by the Hubble Heritage team, please note that the use of this module on your images falls outside of the realm of documentary photography and should preferably noted when publishing your image.



Diffraction partterns are not painted on; they can be quite subtle.

# usage

coming soon

# Wipe: Gradient Removal, Synthetic Flats and Bias correction

The Wipe module detects, models and removes sources of unwanted light bias, whether introduced in the optical train, camera or by light pollution.

The Wipe module upholds StarTools' tradition to solve complex problems with algorithms and data-derived statistics, rather than subjective (and potentially destructive!) manual sample setting and selective processing as found in most other software.

The Wipe module detects, models and removes source of unwanted light bias, whether introduced in the optical train, camera or by light pollution.

# Usage

Wipe is able to detect - and correct for - various complex calibration problems and unwanted artificial signal sources. In addition to a gradient removal routine, it is to detect and model vignetting issues (including over-correction), as well as bias/darks issues.

Common calibration issues include;

- Vignetting manifests itself as the gradual darkening of a dataset towards the corners. It is ideally addressed through flat frame calibration when stacking.
- 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 older DSLRs and Compact Digital Cameras, amp glow often manifests itself as a patch of purple fog near the edge of the image.

Unwanted or artificial signal may include;

- Light pollution, moon glow, airglow, zodiacal light and gegenschein gradients are usually prevalent as gradual increases (or decreases) of background light levels from one corner of the image to another. Most earth-based acquisitions contain a gradient of some form, as even under pristine skies such gradients are prevalent.

- Signal bias is a fixed background levels which, contrary to a gradient, affects the whole image evenly. Most non-normalised datasets exhibit this.
- Amp glow is faint "glow" near one or more edges caused by local thermal noise from heat-dissipating electronics.

While highly effective, it is important to stress that Wipe's capabilities should not be seen as a replacement or long-term alternative to calibrating your datasets with calibration frames; calibrating your dataset with flats, darks and bias masters will always yield superior results. Flats in particular are the #1 way to improve your datasets and the detail you will be able to achieve in your images.

2 sources of unwanted light; a gradient starting at the upper right corner, and light pollution in the form of the typical yellow/brown light. Also visible is vignetting, as seen in the darkening of the corners. Image courtesy of Charles Kuehne.

## Preparing data for the Wipe module

It is of the utmost importance that Wipe is given the best artefact-free, linear data you can muster.

Because Wipe tries to find the true (darkest) background level, any pixel reading that is mistakenly darker than the true background in your image (for example due to dead pixels on the CCD, or a dust speck on the sensor) will cause Wipe to acquire wrong readings for the background. When this happens, Wipe can be seen to "back off" around the area where the anomalous data was detected, resulting in localised patches where gradient (or light pollution) remnants remain. These can often look like halos. Often dark anomalous data can be found at the very centre of

such a halo or remnant.

The reason Wipe backs off is that Wipe (as is the case with most modules in StarTools) refuses to clip your data. Instead Wipe allocates the dynamic range that the dark anomaly needs to display its 'features'. Of course, we don't care about the 'features' of an anomaly and would be happy for Wipe to clip the anomaly if it means the rest of the image will look correct.

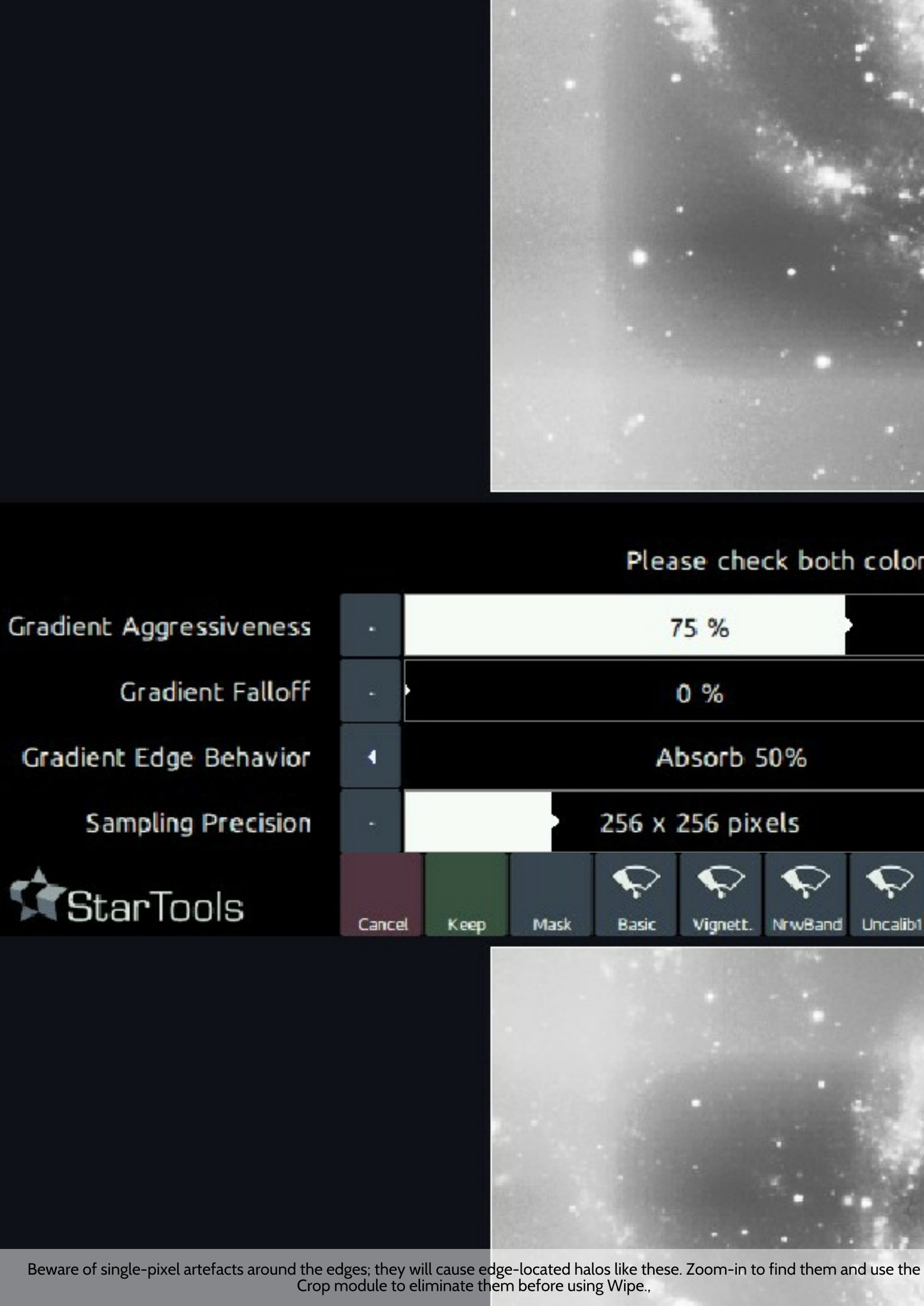
Fortunately, there are various ways to help Wipe avoid anomalous data;

- A '**Dark anomaly filter**' parameter can be set to filter out smaller dark anomalies, such as dead pixels or small clusters of dead pixels, before passing on the image to Wipe for analysis.
- Larger dark anomalies (such as dust specks on the sensor) can be excluded from analysis by, simply by creating a mask that excludes that particular area (for example by ''drawing'' a ''gap'' in the mask using the Lassoo tool in the Mask editor).
- Stacking artefacts should be cropped using the Crop module. Please note that some stackers (e.g. Deep Sky Stacker) can create single column/row pixel stacking artifacts which are easy to miss without zooming in and inspecting the edges of your dataset.

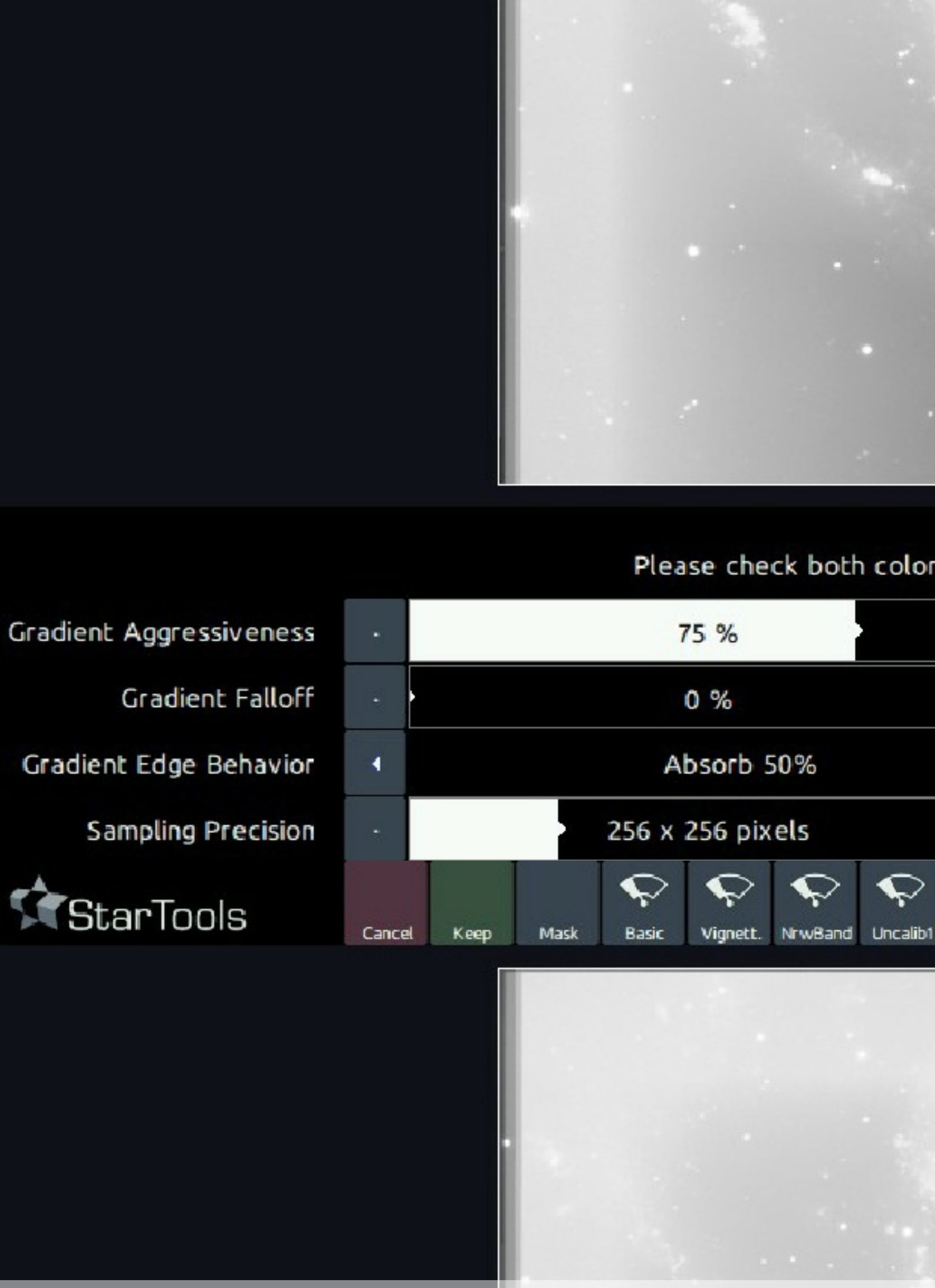
Bright anomalies (such as satellite trails or hot pixels) do not affect Wipe.

## Edge located dark anomalies

Stacking artefacts are the most common dark anomalies located at the edges of your image. Failing to deal with them will lead to a halo effect near the edges of your dataset.



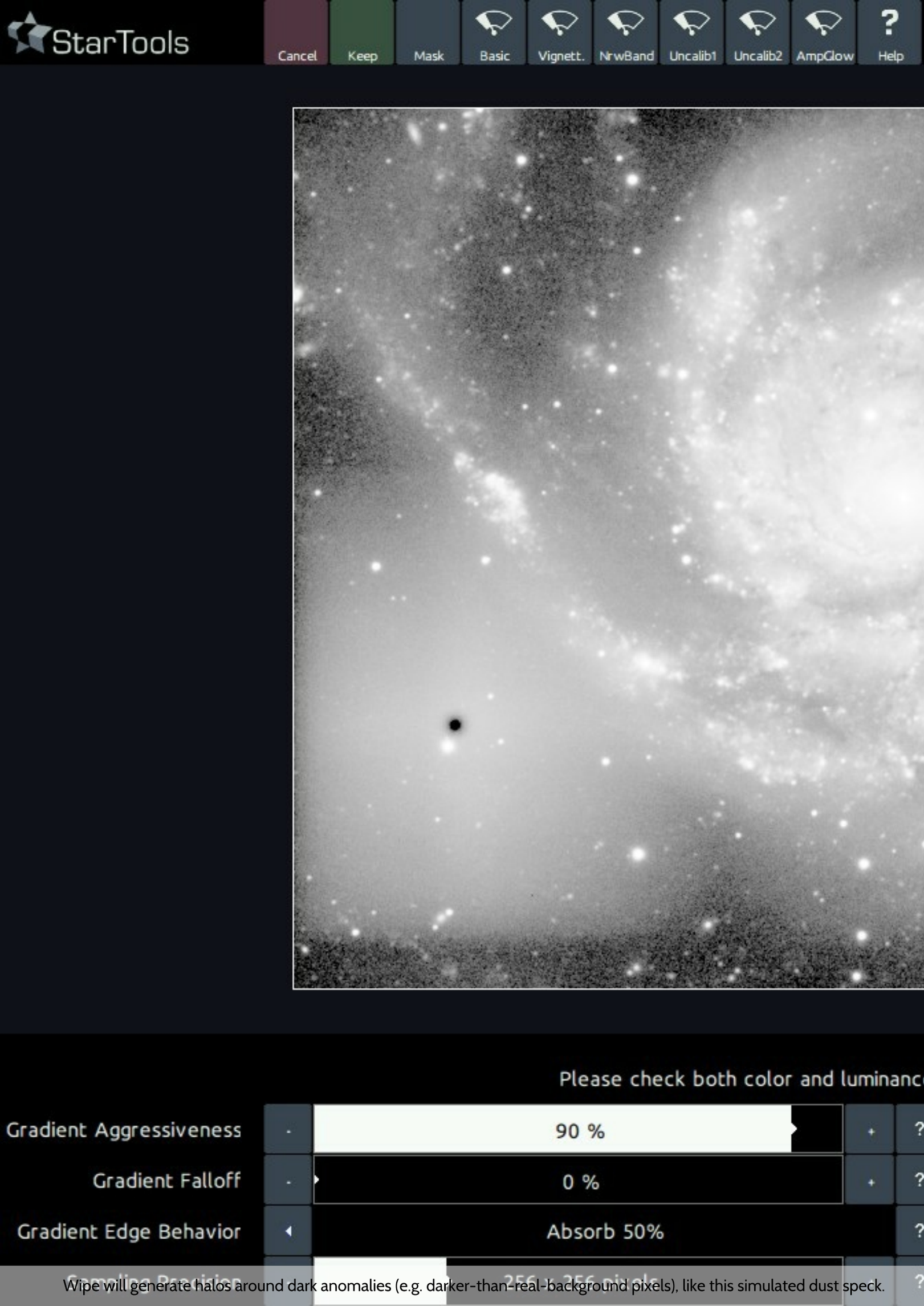

Beware of single-pixel artefacts around the edges; they will cause edge-located halos like these. Zoom-in to find them and use the Crop module to eliminate them before using Wipe.,



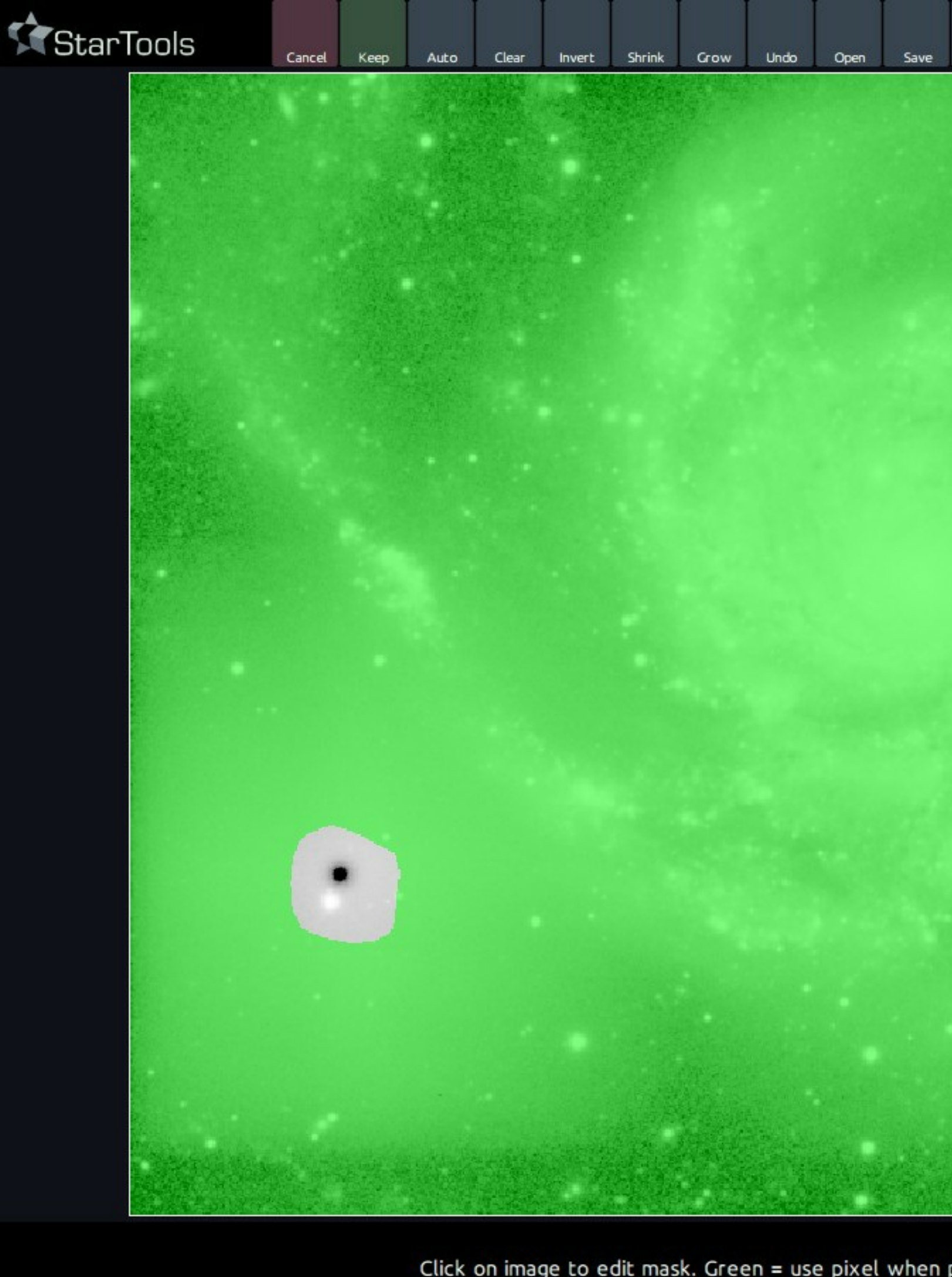

Please remove any stacking artefacts before launching the Wipe module. Failing to do so, will result in edge-loacted halos, like these

## Non-edge located dark anomalies

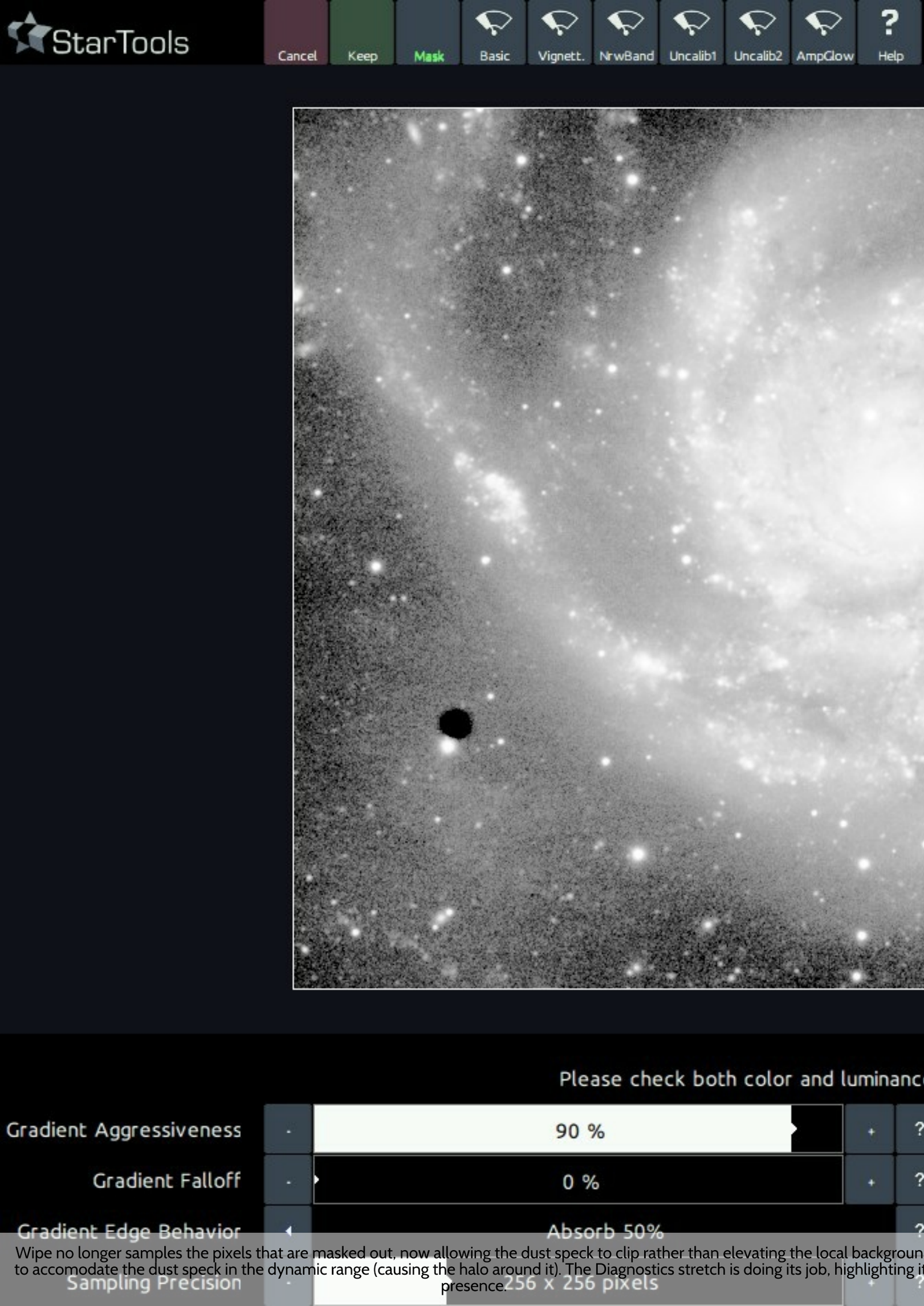
Dust specks, dust donuts, and co-located dead pixels all constitute dark anomalies and will cause halos around them if not taken care of. These type of dark anomalies are taken care of by masking them out so that Wipe will not sample their pixels.



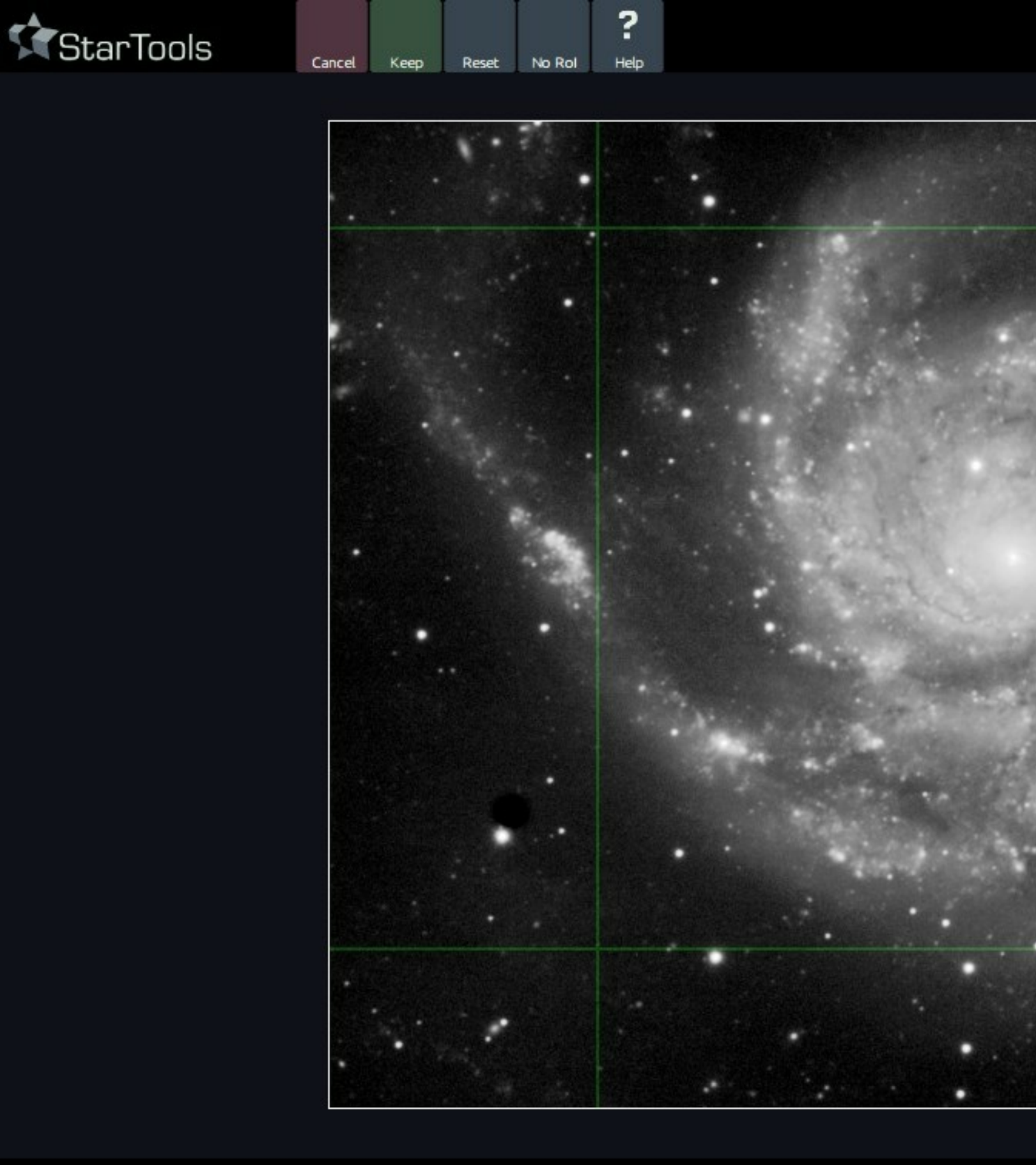

Wipe will generate halos around dark anomalies (e.g. darker-than-real-background pixels), like this simulated dust speck.




In cases where the dark anomaly is too big for the Dark Anomaly Filter parameter to filter out the pixels, you should mask such large dak anomalies out.




Wipe no longer samples the pixels that are masked out, now allowing the dust speck to clip rather than elevating the local backgroun
to accomodate the dust speck in the dynamic range (causing the halo around it). The Diagnostics stretch is doing its job, highlighting it
presence.




A subsequent global stretch in AutoDev makes the dust spec a lot more inconspicuous.

## Operating the Wipe module

Once any dark anomalies in the data have successfully been dealt with, operating the Wipe module is fairly straightforward.

To get started quickly, a number of presets cover some common scenarios;

- '**Basic**' is the default for the Wipe module and configures parameters that work with most well calibrated datasets.
- '**Vignetting**' configures additional settings for vignetting modelling and correction.
- '**Narrowband**' configures Wipe for narrowband datasets which usually only need a light touch due to being less susceptible to visual spectrum light pollution.
- '**Uncalibrated 1**' configures Wipe for completely uncalibrated datasets, for cases where calibration frames such as flats were - for whatever reason - not available. This preset should be used as a last resort.
- '**Uncalibrated 2**' configures Wipe for poor quality, completely uncalibrated datasets. The settings used here are even more aggressive than '**Uncalibrated 1**'. This preset too should only be used as a last resort.

Internally, the module's engine models three stages of calibration similar to an image stacker's calibration stages;

1. synthetic bias/darks modelling and correction (subtraction)
2. synthetic flats modelling and correction (division)
3. gradient modelling and correction (subtraction).

Any issues specified and/or detected are modelled during the correct stage and its results feeds into the next stage.

Synthetic bias/darks modelling

The Wipe module is able to detect horizontal or vertical banding and correct for this. Multiple modelling algorithms are available to detect and mitigate banding.

A defective sensor column repair feature is also available that attempts to recover data that was transposed but not lost, rather than interpolating or 'healing' it using neighbouring pixels.

Synthetic flats modelling

The Wipe module is able to quickly match and model a natural illumination falloff model to your dataset with correction for cropping and off-axis alignment.

Fixed pattern noise and correlated artifact filtering

The '**Correlation Filtering**' parameter specifies the size of correlation artifacts that should be removed. This feature can ameliorate correlation artifacts that are the result of dithering, debayering or fixed pattern sensor cross-talk issues. Correlated noise (often seen as ''worms'', ''clumps'', or hatch-pattern like features) and related artifacts will look like detail to both humans and algorithms. By pre-emptively filtering out these artifacts, modules will be able to better concentrate on the real detail in your dataset and image, rather than attempting to preserve these artifacts.

The usage of this filter is most effective on oversampled data where the artifacts are clearly smaller than the actual resolved detail.

## Gradient modelling and subtraction

Wipe discerns gradient from real detail by estimating undulation frequency. In a nut shell, real detail tends to change rapidly from pixel to pixel, whereas gradients do not. The '**Aggressiveness**' specifies the undulation threshold, whereby higher '**Aggressiveness**' settings latch on to ever faster undulating gradients. At high '**Aggressiveness**' settings, be mindful of Wipe not 'wiping' away any medium to larger scale nebulosity. To Wipe, larger scale nebulosity and a strong undulating gradients can look like the same thing. If you are worried about Wipe removing any larger scale nebulosity, you can designate an area off-limits to its gradient detection algorithm, by means of a mask that masks out that specific area. See the 'Sample revocation' section for more details.

## After Wipe

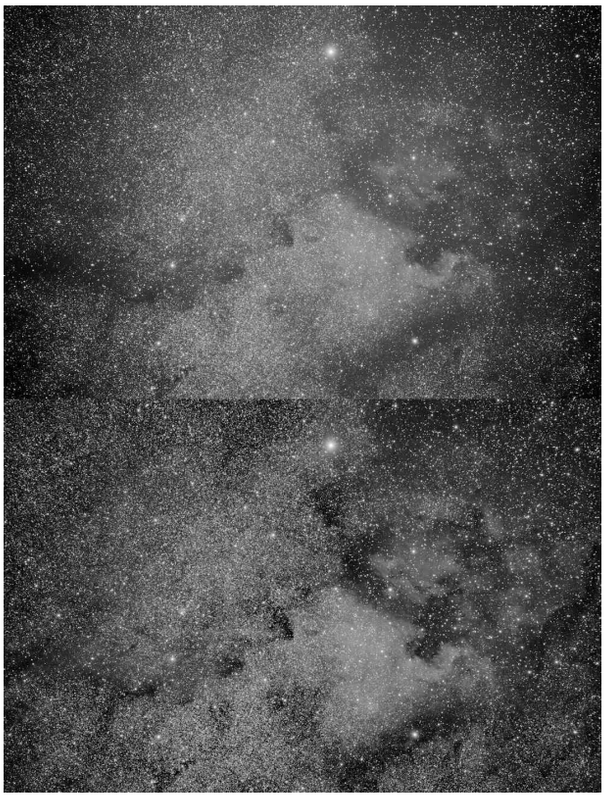
Because Wipe's impact on the dynamic range in the image is typically very, very high, a (new) stretch of the data is almost always needed. This is so that the freed up dynamic range, previously occupied by the gradients,can now be put to good use to show detail. Wipe will return the dataset to its linear state, however with all the cleaning and calibration applied. In essence, this makes a global re-stretch using AutoDev or FilmDev is mandatory after using Wipe. From there, the image is ready for further detail recovery and enhancement, with color calibration preferably done as one of the last steps.

### The diagnostics stretch

Because Wipe operates on the linear data (which is hard to see), a new, temporary automatic non-linear stretch is reapplied on every parameter change, so you can see what the module is doing. The diagnostics stretch is designed to show your dataset in the worst possible light *on purpose*, so you can diagnose issues and remedy them. The sole purpose of this stretch is to bring out any latent issues such as gradient, dust donuts, dark pixels. That is, it is entirely meant for diagnostics purposes inside the Wipe module and in no way, shape or form should be regarded as a suggested final global stretch.

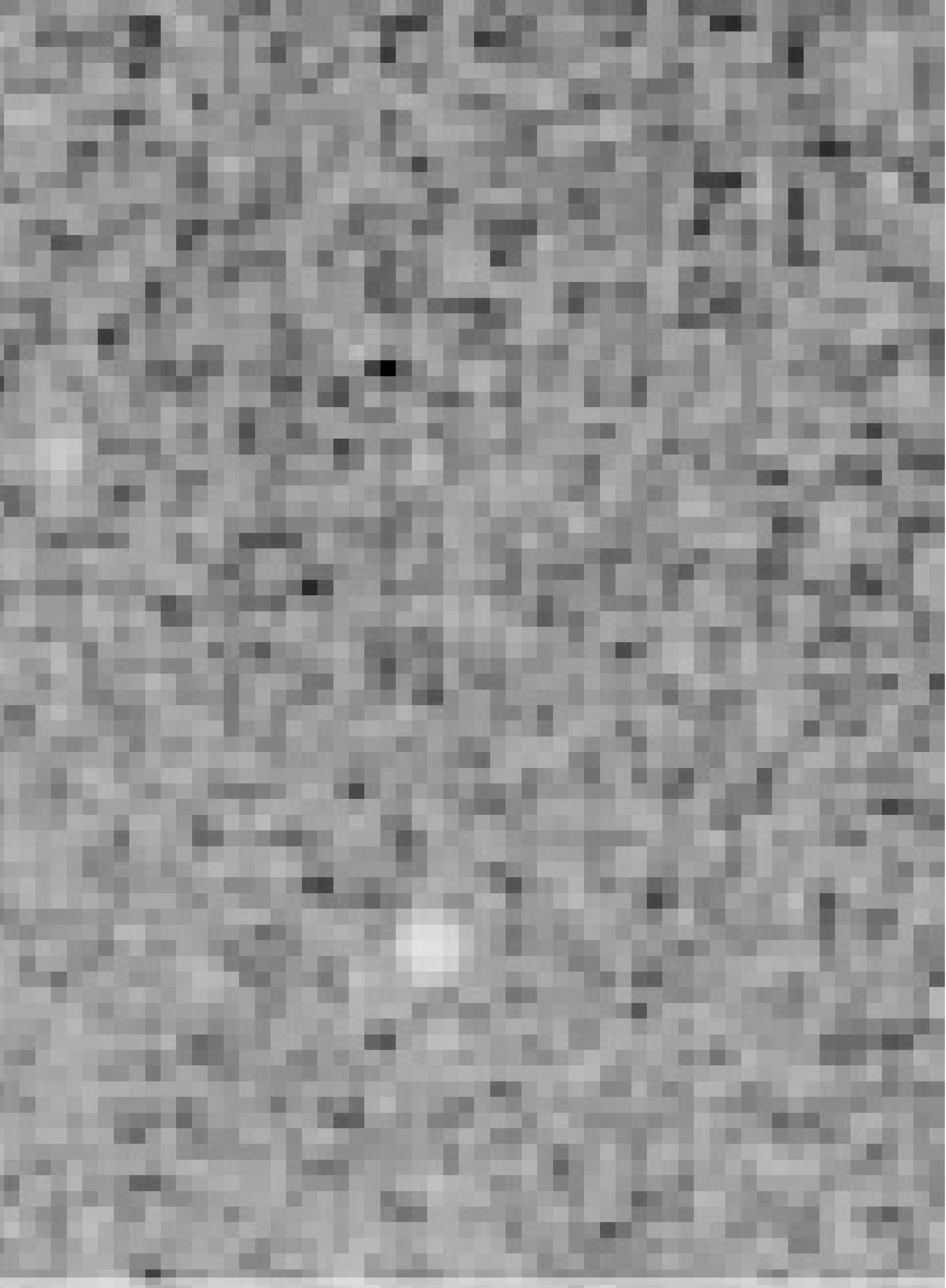
### Automatically separated luminance and chrominance datasets

If Compose mode is engaged (see Compose module), Wipe processes luminance (detail) and chrominance (colour) separately, yet simultaneously. If you process in Compose mode (which is recommended), you should check both the results for the luminance and chrominance portion of your image. Before keeping the result, the Wipe module will alert you to this once, if you have not done so.



The 'Uncalibrated' presets model - and correct for - vignetting, as well as gradients.

Wipe, as part of the three stages of operation, is able to recover data from defective sensor rows and columns.



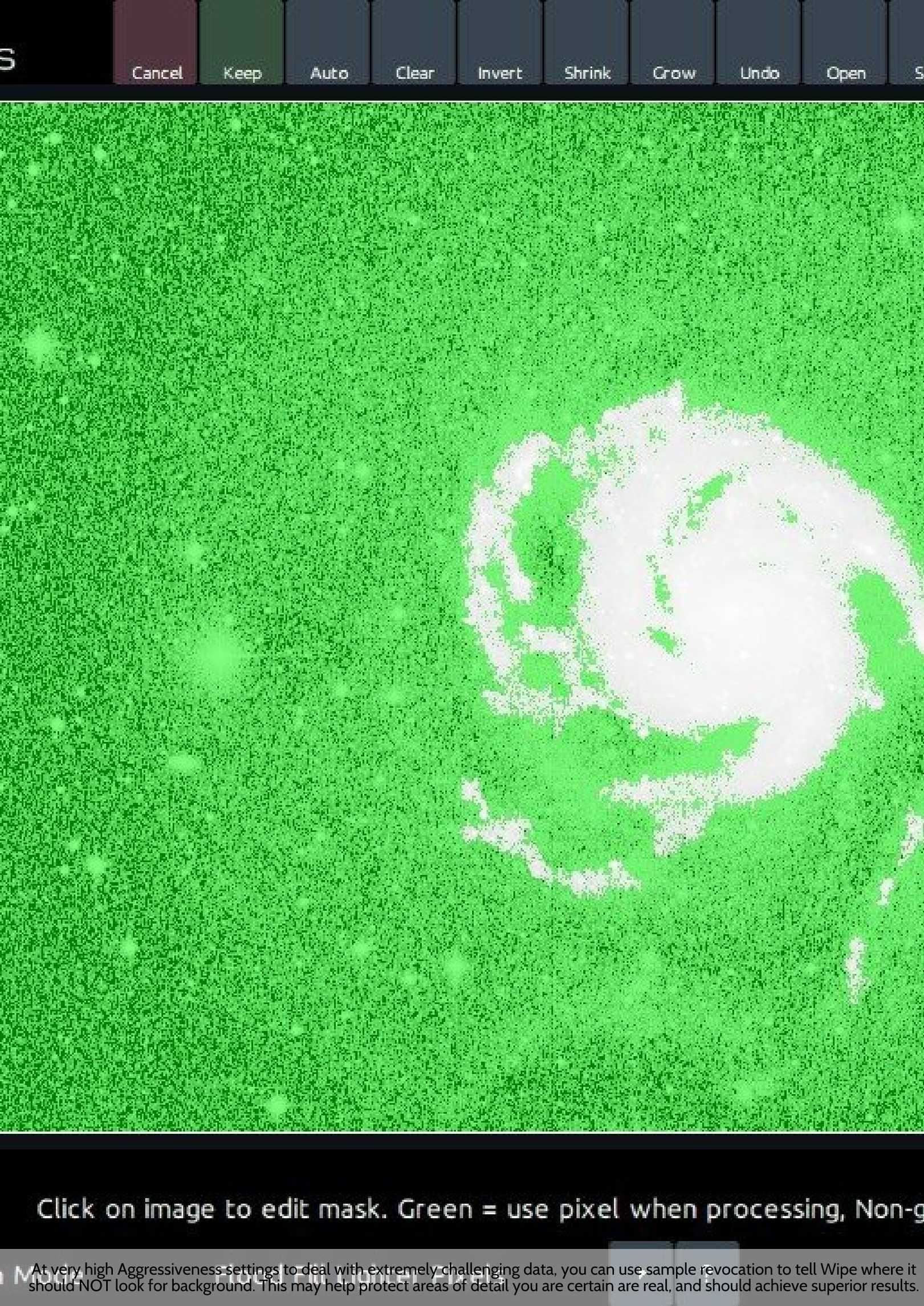
The Correlation Filtering parameter can ameliorate correlation artifacts that are the result of dithering, debayering or fixed pattern sensor cross-talk issues.

## Sample revocation

With the exception of the previously mentioned larger "dark anomalies" (such as dust donuts or clumps of dead pixels), it is typically unnecessary to provide Wipe with a mask. However if you wish to give Wipe specific guidance, with respect to which areas of the image to include in the model of the background, then you may do so with a mask that describes where background *definitely does not exist*.

This is a subtle but important distinction from background extraction routines in less sophisticated software, where the user must "guess" where background *definitely exists*. The former is easy to determine and is readily visible, whereas the latter is usually impossible to see, precisely because the background is mired in gradients. In other words, StarTools' Wipe module works by *sample revocation* ("definitely nothing to see here"), rather than by the less optimal (and possibly destructive!) *sample setting* ("there is background here").

Analogous to how sample setting routines yield poor results by accidentally including areas of faint nebulosity, the opposite is the case in Wipe; accidentally masking *out* real background will yield the poorer results in Wipe. Therefore, try to be conservative with what is being masked out. If in doubt, leave an area masked *in* for Wipe to analyse.




At very high Aggressiveness settings to deal with extremely challenging data, you can use sample revocation to tell Wipe where it should NOT look for background. This may help protect areas of detail you are certain are real, and should achieve superior results.

## Design philosophy and limitations

As with all modules in StarTools, the Wipe module is designed around robust data analysis and *algorithmic* reconstruction principles. The data should speak for themselves and manual touch-ups or subjective gradient model construction by means of sample setting is, by default, avoided as much as possible.

In general, StarTools' Wipe module should yield superior results, retaining more faint detail and subtle large-scale nebulosity, compared to basic, traditional manual gradient model construction routines. However, exceptions arise where gradients undulate (e.g. rise or fall) faster than the detail in the image due to atypical acquisition issues (incorrect flat frames, very strongly delineated localised light pollution domes). Human nor machine will be able to discern detail objectively or with certainty. As a result Wipe will, likewise, struggle in such cases.

StarTools

Advanced image processing software for astrophotography

SiliconFields

www.startools.org

startoolsastro@gmail.com

ABN 42 373 023 390

© 2024 SiliconFields

Download FREE trial

https://www.startools.org/downloads/