StarTools

# Signal evolution Tracking

# The one thing you should know about StarTools

It's a feature called "Tracking" and processes your signal in temporal / 3D (X, Y, t) space, rather than standard 2D (X,Y) space.

The result is less noise grain, finer detail, more flexibility, and unique functionality. You will not find this in any other software.

# The best kept secret amongst signal processing purists

StarTools monitors your signal and its noise component, *per-pixel*, throughout your processing (time). It sports image quality and unique functionality that *far* surpasses other software. Big claim? Let us back it up.

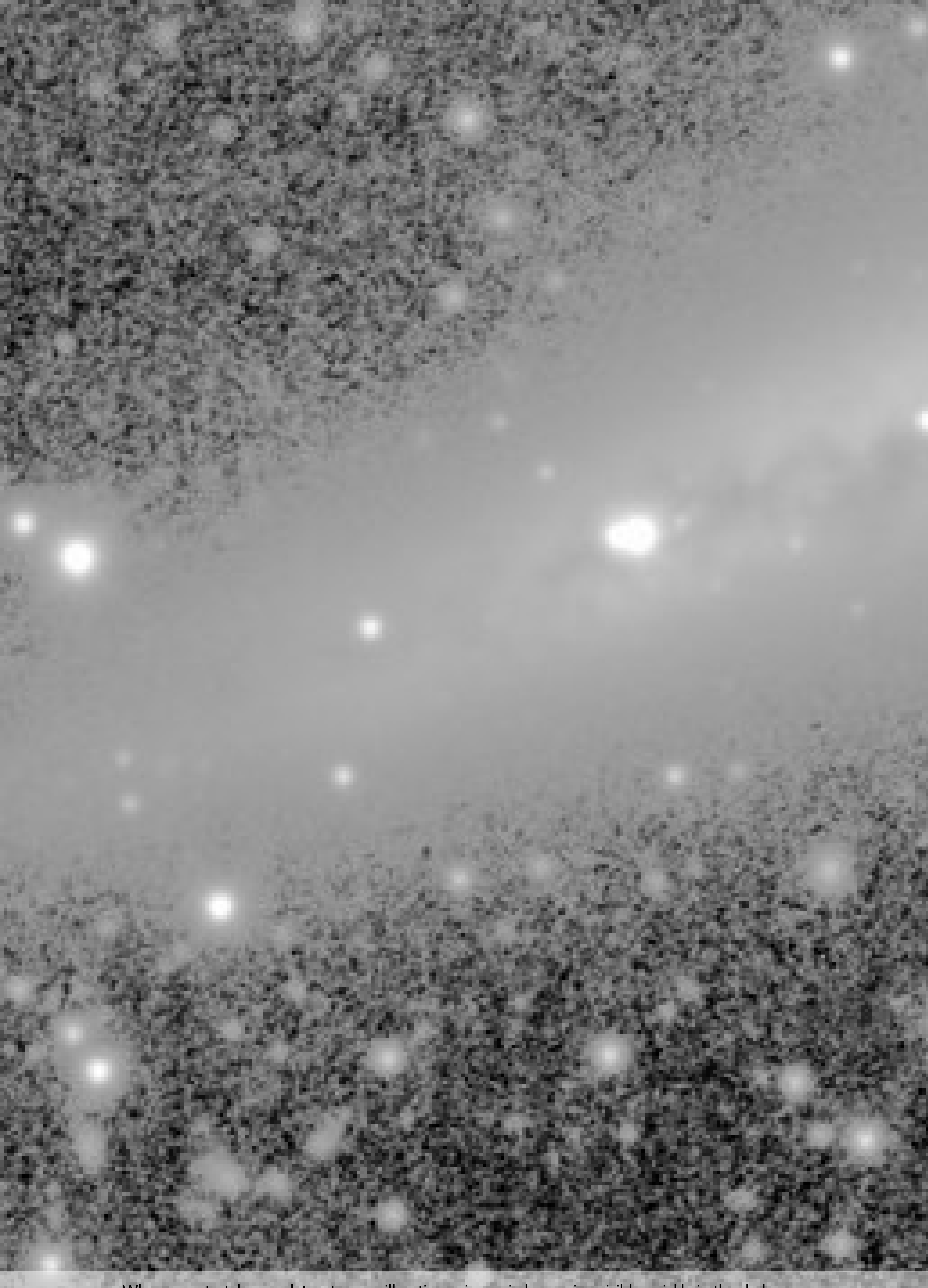
## Your signal and its noise component

If you have ever processed an astro image, you will have had to non-linearly stretch the image at some point, to make the darker parts with faint signal visible. Whether you used levels & curves, digital development, or some other tool, you will have noticed noise grain becoming visible quickly.

You may have also noticed that the noise grain always seems to be worse in the darker areas than the in brighter areas. The reason is simple; when you stretch the image to bring out the darker signal, you are also stretching the noise component of the signal along with it.

And the former is just a simple global stretch. Now consider that every pixel's noise component goes through many other transformations and changes as you process your image. Once you get into the more esoteric and advanced operations such as local contrast enhancements or wavelet sharpening, noise levels get distorted in all sorts of different ways in all sorts of different places.

The result? In your final image, noise is worse in some areas, less in others. A "one-noise-reduction-pass-fits-all" no longer applies. Yet that's all other software packages - even the big names - offer. Why? Because tracking that noise grain evolution across your *full* workflow is very, very hard to implement.



When you stretch your dataset, you will notice noise grain becoming visible quickly in the darker areas..

# Traditional image processing software for astrophotography is fundamentally broken

Chances are you have used noise reduction at some stage. In astrophotography, the problem with most noise reduction routines, is that they have no idea *how much* worse the noise grain has become (or *will* become) in your image as you process(ed) it. These routines, have no idea how you stretched and processed your image earlier or how you will in the future. And they certainly have no idea how you squashed and stretched the noise component locally with wavelet sharpening or local contrast optimisation.

In short, the big problem, is that separate image processing routines and filters have no idea what came before, nor what will come after when you invoke them. All pixels are treated the same, regardless of their history (is this pixel from a high SNR area or a low SNR area? Who knows?). Current image processing routines and filters are still as 'dumb' as they were in the early 90s. It's still ''input, output, next''. They pick a point in time, look at the signal and estimated noise component and do their thing. This is still true for black-box AI-based algorithms; they cannot predict the future.

Without knowing how signal and its noise component evolved to become your final image, trying to, for example, squash *visual* noise accurately is fundamentally impossible. What's too much in one area, is too little in another, all because of the way prior filters have modified the noise component beforehand. The same is true for applying noise reduction before stretching (e.g. at the linear stage); noise grain is ultimately only a problem when it becomes visible, but at the linear stage this hasn't happened yet. The only reason then to apply any noise reduction at the linear stage, is if your software's algorithms cannot cope with noise effectively; and that is a poor reason for destroying (or blatantly inventing) signal so early on.

The separation of image processing into dumb filters and objects, is one of the biggest problems for signal fidelity in astrophotographical image processing software today. It is the sole reason for poorer final images, with steeper learning curves than are necessary. Without addressing this fundamental problem, "having more control with more filters and tools" is an illusion. The IKEA effect aside, long workflows with endless tweaking and corrections do not make for better images. On the contrary, they make for much poorer images, or do no longer reflect a photographic reality.

Now imagine every tool, every filter, every algorithm could work backwards from the finished image, tracing signal evolution, per-pixel, all the way back to the source signal? That's Tracking!

# Tracking is the solution

## Tracking is time travel

Tracking in StarTools makes sure that every module and algorithm can trace back how a pixel was modified at any point in time. It is the Tracking engine's job to allow modules and algorithms ''travel in time'' to consult data and even change data (changing the past), and then forward-propagate the changes to the present.

The latter sees the Tracking module re-apply every operation made since that point in time, however with the changed data as a starting point; changing the past for a better future. This is effectively signal processing in three dimensions; X, Y and time (X, Y, t).

# The power of temporal 3D (X, Y, t) signal processing

## Deconvolution; an example

This remarkable feature is responsible for never-seen-before functionality that allows you to, for example, apply correct deconvolution to heavily processed data. The deconvolution module "simply" travels back in time to a point where the data was still linear (deconvolution can only correctly be applied to linear data!). Once travelled back in time, deconvolution is applied and then Tracking forward-propagates the changes. The result is exactly what your processed data would have looked like with if you had applied deconvolution earlier and then processed it further.

Sequence doesn't matter any more, allowing you to process and evaluate your image as you see fit. But wait, there's more!

## Deconvolution; an example that gets even better

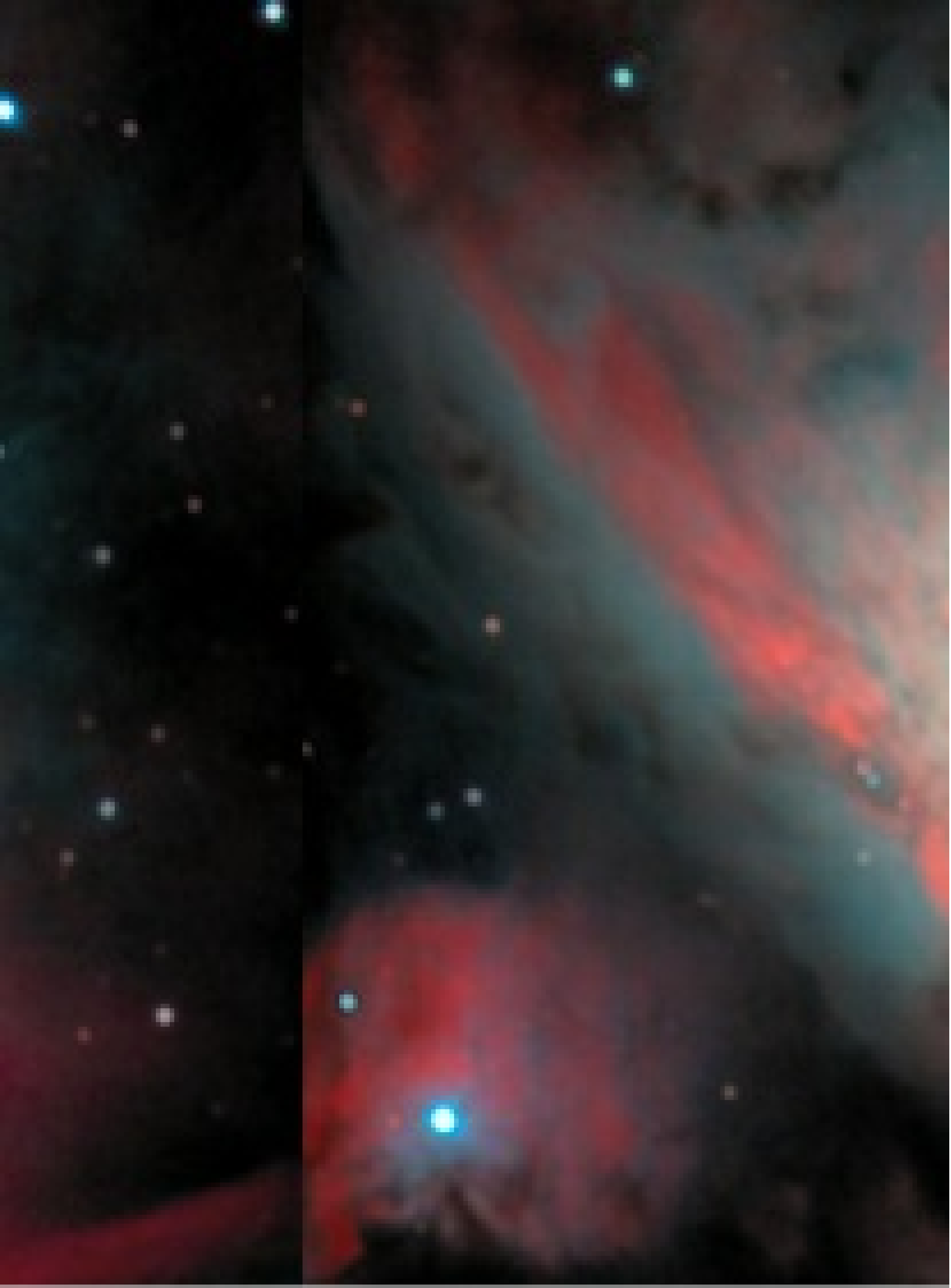
Time travelling like this is very useful and amazing in its own right, but there is another *major difference* in StarTools' deconvolution module;

Because you initiated deconvolution at a later stage than *normally* can be the case, the deconvolution module can take into account how you further processed the image after it normally should have been invoked. The deconvolution module now has knowledge about a future it normally is not privy to in any other software. Specifically, that knowledge of the future, tells it exactly how you stretched and modified every pixel - including its noise component - after the time its job *should* have been done.

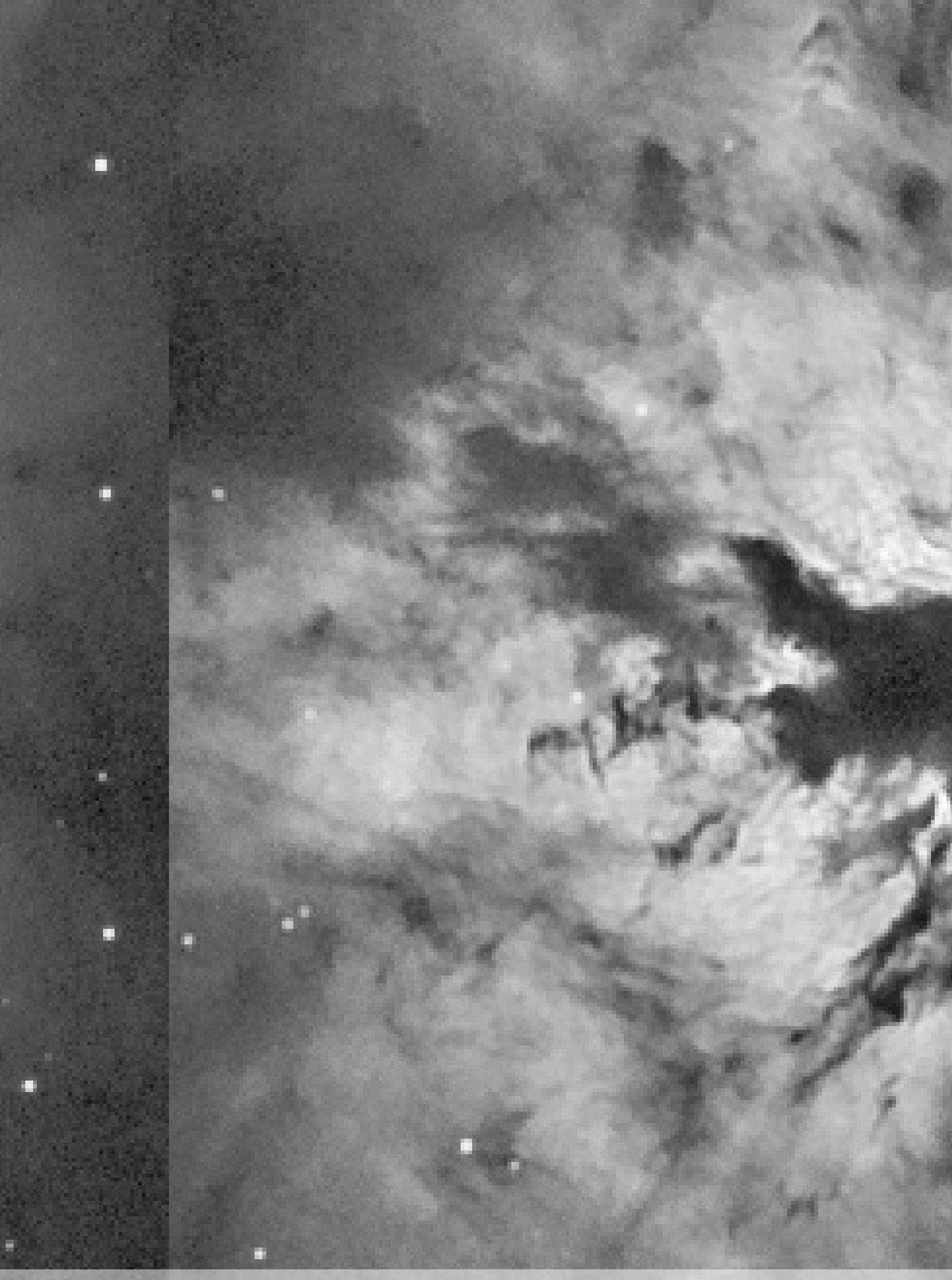
You know what really loves per-pixel noise component statistics like these? Deconvolution regularization algorithms! A regularization algorithm suppresses the creation of artefacts caused by the deconvolution of - you guessed it - noise grain. Now that the deconvolution algorithm knows how noise grain will propagate in the "future", it can take that into account when applying deconvolution at the time when your data is still linear, thereby avoiding a grainy "future", while allowing you to gain more detail. It is like going back in time and telling yourself the lottery numbers to today's draw.

What does this look like in practice? It looks like a deconvolution routine that just "magically" brings into focus what it can. No sub-optimal local supports needed, no subjective luminance masks needed, no selective blending needed. There is no exaggerated noise grain, just enhanced detail; objectively better results, in less time, with less hassle.

And all this is just what Tracking does for the deconvolution module. There are many more modules that rely on Tracking in a similar manner, achieving objectively better results than any other software, simply by being smarter - much smarter - with your hard-won signal. *This* is what StarTools is all about.



Signal evolution Tracking allows for many more enhancements over traditional software. For example color constancy (right), effortlessly visualizing features with similar chemical/phsycial properties, regardless of brightness.



200% zoomed crop input and output. Left: original, right: Decon result. Thanks to signal evolution Tracking, and despite stretching, local dynamic range optimization and noise presence, Decon is able to recover the finest details without introducing artifacts, or the need for support masks or manual intervention.

# Mathematical implementation of the signal evolution Tracking engine

In conventional processing engines, every pixel as-you-see-it is the result of the operation *that was last carried out* (some simple screen stretch capabilities excepted to visualise linear data). Operations are carried out one after the other and exist in some linear stack (typically accessible via an 'undo' history). The *individual operations* however, have no concept of what other operations preceded them, nor what operations will follow them, nor what the result was or will be. Signal flows one way in time; forward. Conventional software does not feed back signal, nor propagates it back and forth in order to refine the final result of the stack or 'undo' history.

Some software platforms even mistakenly implement astronomical signal processing in a formalised *object oriented* platform. An object oriented approach, by definition, implements strict decoupling of the individual operations, and formalises complete unawareness of the algorithms contained therein, with regards to *where* and *when* in the signal flow they are being invoked. This design completely destroys any ability of such algorithms to know what augmenting data or statistics may be available to them to do a better job. Worse, such software allows for entirely nonsensical signal flows that violate mathematical principles and the physics these principles are meant to model. The result is lower quality images through less sophisticated (but more numerous) algorithms, rounding errors, user-induced correction feedback loops (invoking another module to correct the output of the last), and steeper learning curves than necessary.

In contrast, StarTools works by constantly re-building and refining a single equation, for every pixel, that transforms the source data into the image-as-you-see-it. It means there is no concept of linear versus non-linear processing, there are no screen stretches with lookup tables, there is no scope for illegal sequences, there is no overcooking or noise grain/artefact propagation, there are no rounding errors. What you see is the shortest, purest transformation of your linear signal into a final image. And what you see is what you get.

Even more ground-breaking; substituting some of its variables for the equation itself (or parts thereof), allows complex feedback of signal to occur. This effectively provides, for example, standard algorithms like deconvolution or noise reduction, precise knowledge about a "future" or "past" of the signal. Such algorithms will be able to accurately calculate how the other algorithms will behave in response to their actions anywhere on the timeline. The result is that such algorithms are augmented with comprehensive signal evolution statistics and intelligence *for the user's entire workflow.* This lets these algorithms yield greatly superior results to that of equivalent algorithms in conventional software. Applying the latter innovation to - otherwise - standard, well known algorithms is, in fact, the subject of most of StarTools' research and development efforts.

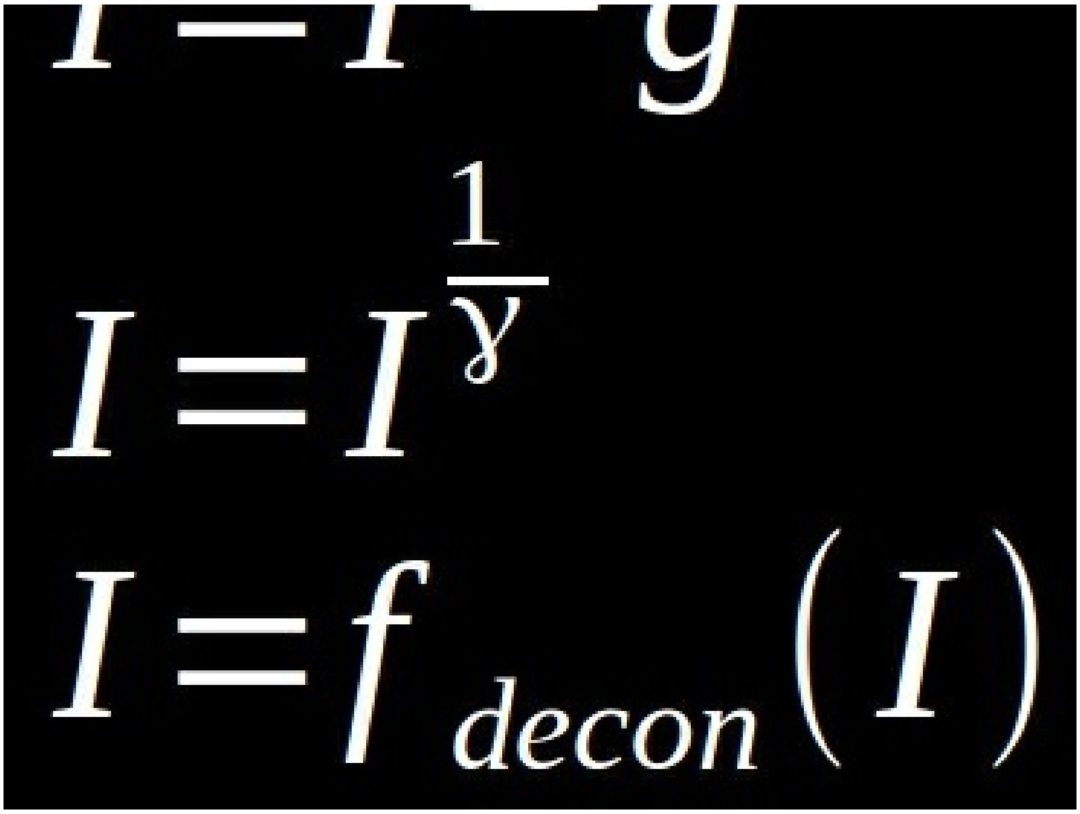
$$I^{\gamma} \qquad I =$$

$$I - g \qquad I =$$

$$f_{decon}(I) \; I =$$

Image processing in a traditional application; every step uses the output of the previous step as its input. This workflow shows a sequence where the user subtracts a gradient "g", performs deconvolution on the linear data and then performs a non-linear gamma correction stretch.

The power of StarTools' novel engine, is not only expressed in higher signal fidelity and lifting of limitations of conventional engines; its power is also expressed in *ease-of-use*. Illegal or mathematically incongruent paths are closed off, while parameter tweaks always yield useful and predictable results. Defaults just work for most datasets, proving that the new engine is universally applicable, consistent and rooted in a mathematically sound signal processing paradigm.



$$I = I^{\frac{1}{\gamma}}$$

$$I = f_{decon}(I)$$

An "illegal" sequence where deconvolution is erroneously applied after a non-linear gamma correction stretch.

$$_{gin} - g$$

$$_{rigin} - g\Big)^{\frac{1}{\gamma}}$$

$$econ\Big(I_{origin}$$

In StarTools, your image as-you-see-it is the result of an ever changing equation that is mathematically sound and simplified. This equation shows a sequence where the user subtracts a gradient "g", performs a gamma correction, and performs deconvolution (noting that the deconvolution step is normally considered out-of-sequence). As a result of the equation building, non-linear vs linea procesing is completely abstracted away.

An algorithmic tour de force; enhancing a standard Richardson & Lucy deconvolution iteration by basing the regularization step on a forward propagated version of the previous iteration, thereby taking into account artefact propagation in the user's full workflow in th stretched, processed domain; this makes an algorithm like deconvolution privy to statistics that are far outside its normal linear-only purview. Substitution and equation solving like this, encompassing the entire workflow, is impossible in convential software.

# Do it once, do it right

Physics and applied mathematics demand that some operations are done in a particular order. No ifs, no buts. Certain operations have one specific place in a sound signal flow, yet others have less rigid sequence requirements. Whichever your processing decisions, they are worked into the equation in a mathematically congruent way.

The most elegant equation is often the shortest one. In StarTools, you refine the final equation like a sculptor would refine a coarse piece of marble into a sculpture; from coarse gestures to fine tweaks. Module functionality does not overlap in StarTools; you will never be correcting one module's output with another module that does-the-same-thing-but-differently. I.e. the engine's goal is to "tack on" to the equation as little as possible, and to rather tweak its present form and variables as much as possible.

Less is more. The shorter solution is the better solution. The best part is no part. Endless tweaking is not a thing in StarTools, and all decisions and module invocations are meant to be done with a clear direction, decisiveness and purpose. Feeling a sense of closure on an image is a feature, not a bug.

A good example of the "*do it once, do it right*" philosophy that StarTools' engine affords, is its approach to noise reduction. In StarTools you don't need to "help" any of the algorithms by noise reducing earlier in your workflow and passing them noise reduced versions of your datasets. All modules are fully noise-aware. As such, in StarTools, noise is an aesthetic consideration only. Noise grain only becomes a problem if it is visible and aesthetically objectionable. Therefore noise reduction is only applied at the very last moment, when it is at its *most* visible and *most* objectionable. In StarTools, you should never apply noise reduction to an *unfinished* image; any further processing will change your image's noise profile again, invalidating your previous noise reduction decisions and efforts. As such, there is only *one* noise reduction tool and *one* noise reduction moment; the one *right* tool at the one *right* moment. That is, a tool that models the noise profile in your image with pin-point accuracy, at the very end of your workflow.

## No AI-based neural hallucination that invents "plausible" signal

StarTools prides itself on robustly implementing physics-based algorithms, as well as documentary fidelity.

StarTools does not encourage, nor enable practices that introduce unquantifiable "make believe" signal, without transparently warning the user of the consequences of exterminating documentary fidelity. Usage of unsophisticated algorithms that use basic neural hallucination in response to an impulse, have no place in documentary astrophotography; they invariably introduce signal that was never recorded (documented).

StarTools' principal developer is not exactly a Luddite when it comes to AI - he studied it and is named inventor on a number of AI-related patents. More than most, he is excited by how AI improves our daily lives, from augmented medical diagnoses to self-driving cars. The future of AI - overall - is an incredibly bright one.

## "I reject your reality and substitute my own"

The flipside of AI is that it can be used for deception, whether borne out of ignorance, insecurities or malice. Neural hallucination - the lowest hanging AI fruit - is quite literally not a substitute for real recorded detail. Just like educated guesses are not a substitute for real measurements.

Just like most scoff at applying an AI Instagram filter and passing it off as a real photo of a person, so should an documentary astrophotographer scoff at applying an AI "Instagram filter" to their data and pass it off as something that was recorded for real.

StarTools will not ever sell you on "game changing" snake oil or open up its platform for other actors to do the same. In honest, documentary astrophotography, the game is about turning signal into detail in the face of noise. We choose to focus our development efforts on giving you the best odds in that game, but we will never help your rig it.

## TL;DR

In StarTools, your signal is processed (read and written) in a time-fluid way, by means of an ever changing equation for every pixel. Being able to change the past for a better future not only gives you amazing new functionality, changing the past with knowledge of the future also means a much cleaner signal. Tracking always knows how to accurately estimate the noise component in your signal, no matter how heavily modified. Unnecessary subjectivity, sub-optimal sequences and overcooking are - literally - taken out of the equation, yielding vastly better results in less time.

# User-friendly by mathematical nature

For its unique engine to function, StarTools needs to be able to make mathematical sense of your signal flow. That's why it's simply unable to perform "nonsensical" operations. This is great if you're a beginner and saves you from bad habits or sub-optimal decisions.

Just like in real life, in astrophotographical image processing, some things need to be done in a particular order to get the correct result. Folding, drying then washing your shirt, will achieve a markedly different result to washing, drying and folding it. Similarly, deconvolution will not achieve correct results if it is done after stretching, ditto for light pollution removal and color calibration. In mathematics, this is called the commutative property.

The "Tracking" feature, constantly backward propagates and forward propagates your signal through processing "time" as needed. This means that "nonsensical" signal paths (e.g. signal paths that get sequences wrong) would break Tracking's ability. Therefore, such signal paths are closed off. For this reason, it is neigh-impossible in StarTools to perform catastrophically destructive operations on your data; it simply wouldn't be sound mathematics and the code would break.

For example, the notion of processing in the linear domain vs non-linear (stretched) domain is completely abstracted away by the engine because it *needs* to do that. If you didn't know the difference between those two yet, you can get away with learning about this later. Even without knowing the ins-and-outs of astronomical signal processing, you can still produce great images from the get-go; StarTools takes care of the correct sequence.

So, whereas other software will happily (and incorrectly!) allow you to perform light pollution removal, color calibration or deconvolution after stretching, StarTools will...

***...actually also let you do that, but with a twist!***

Tracking will rewind and/or fast-forward to the right point in time, so that the signal flow to makes sense and is mathematically consistent. It inserts the operation in the correct order and recalculates what the result would have looked like if your decision had always been the case. It's time travelling for image processing, where you can change the past to affect the present and future.

For an in-depth explanation of Tracking, see the Tracking section.

StarTools

Advanced image processing software for astrophotography

SiliconFields

www.startools.org

startoolsastro@gmail.com

ABN 42 373 023 390