

Noise Reduction and Sharpening

Astrophotography requires specialized techniques to reduce image noise and improve definition, without each destroying the other.

On our journey through critical PixInsight processes, noise reduction and sharpening are the next step. These are two sides of the same coin; sharpening often makes image noise more obvious and noise reduction often reduces the apparent sharpness. Considering them both at the same time makes sense, as the optimum adjustment is always a balance between the two.

Both processes employ unique tools but share some too. They are often deployed a little during linear and non-linear processing for best effect. These are uniquely applied to different parts of the image to achieve the right balance between sharpening and noise reduction, rather than to the image as a whole, and as a result, both noise reduction and sharpening are usually applied through a mask of some sort. Interestingly, a search of the PI forum for advice on which to apply first, suggests that there are few rules. One thing is true though, stretching and sharpening make image noise more obvious and more difficult to remove. The trick, as always, is to apply manipulations that do not create more issues than they solve.

Both noise reduction and sharpening techniques affect the variation between neighboring pixels. These can be direct neighbors or pixels in the wider neighborhood, depending on the type and scale of the operation. Noise can be measured mathematically but sharpness is more difficult to assess and relies upon judgement. Some texts suggest that the Modulation Transfer Function (MTF) of a low-resolution target is a good indicator of sharpness, in photographic terms using the 10 line-pairs/mm transfer function (whereas resolution is indicated by the contrast level of 40 line-pairs/mm). In practice, however, there is no precise boundary between improving the appearance of smaller structures and enhancing the contrast of larger ones. As such, there is some overlap between sharpening and general local contrast enhancing (stretching) tools, which are the subject of the next chapter.

Typically noise reduction concepts include:

- blurring; reducing local contrast by averaging a group of neighboring pixels, affecting all pixels
- selective substitution; by replacing outlier pixels with an aggregate (for example, median) of its surroundings

Sharpening includes these concepts:

- deconvolution (explained in its own chapter)
- increasing the contrast of small-scale features
- enhancing edge contrasts (the equivalent of acutance in film development)
- more edge effects such as unsharp mask (again, originating from the traditional photographic era)

The marriage of the two processes is cemented by the fact that, in some cases, the same tool can sharpen and reduce noise. In both processes we also rely upon the quirks of human vision to convince oneself that we have increased sharpness and reduced noise. Our ability to discern small changes in luminosity diminishes with intensity and as a result, if we were to compare similar levels of noise in shadow and mid-tone areas, we would perceive more noise in the mid-tone area. Our color discrimination is not uniform either and we are more sensitive to subtle changes in green coloration, which explains why color cameras have two green-filtered photosites for each red and blue in the Bayer array. There are a few other things to keep in mind:

- Noise and signal to noise ratio are different: Amplifying an image does not change the signal to noise ratio but it does increase the noise level. Noise is more apparent if the overall signal level is increased from a shadow level to a mid-tone.
- Non-linear stretches may affect the signal to noise ratio slightly, as the amplification (gain) is not applied uniformly across different image intensities.
- Noise levels in an image are often dominated by read noise and sky noise – both of which have uniform levels. The signal to noise ratio, however, will be very different between bright and dark areas in the image. Brighter areas can withstand more sharpening and require less noise reduction.
- The eye is adept at detecting adjacent differences in brightness. As a consequence, sharpening is mostly applied to the luminance data.
- Sharpening increases contrast and may cause clipping and/or artefacts.
- Some objects do not have distinct boundaries and sometimes, as a consequence, less is more.

- The stretching process accentuates problems, so be careful and do not introduce subtle artefacts when sharpening or reducing noise in a linear image.

Noise Reduction

The available tools in PixInsight have changed over the last few years and as they have been updated, a few have fallen by the wayside. It is a tough old world and image processing is no exception. So, if you are looking for instruction on using AdaptiveContrast-DrivenNoise Reduction (ACDNR) or AtrousWaveletTransform (ATWT), there is good and bad news; they have been moved to the obsolete category but have more effective replacements in the form of MultiscaleLinearTransform (MLT), MultiscaleMedianTransform (MMT), TGVDenoise and some very clever scripts. Before looking at each, in turn, we need to consider a few more things:

- Where in the workflow should we reduce noise and by how much (and at what scale)?
- How do we protect stars?
- How do we preserve image detail?
- How do we treat color (chroma) noise?

As usual, there are no hard and fast rules but only general recommendations; the latest noise reduction techniques are best applied before any image sharpening (including deconvolution) and yet, it is also often the case that a small dose of sharpening and noise reduction prior to publication is required to tune the final image. The blurring effect of noise reduction potentially robs essential detail from an image and it is essential to ensure that either the tool itself or a protection mask directs the noise reduction to the lowest SNR areas and equally does not soften star boundaries. Excessive application can make backgrounds look plastic and the best practice is to acquire sufficient exposure in the first place and apply the minimum amount of noise reduction on the linear, integrated image before any sharpening process.

MureDenoise Script

This tool has evaded many for some time as it is hidden in the PixInsight script menu. It works exclusively on linear monochrome images (or averaged combinations) corrupted by shot, dark current and read noise. Its acronym is a tenuous contrivance of an “interscale wavelet Mixed noise Unbiased Risk Estimator”. Thankfully it works brilliantly on image stacks, especially before any sharpening, including deconvolution. One of its attractions is that it is based on sensor parameters and does not require extensive tweaking. The nearest performing equivalent,

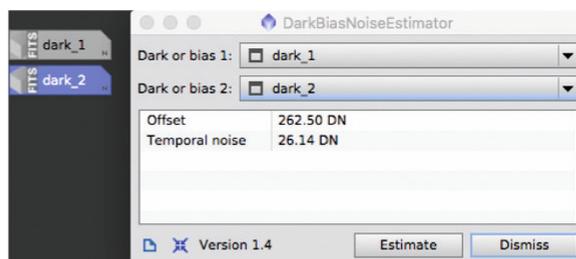


fig.1 A good estimate of Gaussian noise for the MureDenoise script is to use the Temporal noise assessment from two dark or bias frames in the DarkBiasNoiseEstimator script.

and only after extensive trial and error, is the Multiscale Linear Transformation tool.

In use, the tool requires a minimum of information (fig.2) with which to calculate and remove the noise. This includes the number of images in the stack, the interpolation method used by image registration and the camera gain and noise. The last two are normally available from the manufacturer but can be measured by running the FlatSNREstimator and DarkBiasNoiseEstimator scripts respectively on a couple of representative flat and dark frames. An example of using the DarkBiasNoiseEstimator script to calculate noise is shown in fig.1. The unit DN refers to a 16-bit data number (aka ADU), so, in the case of a camera with 8e read (Gaussian) noise and a gain of 0.5e / ADU, the Gaussian noise is 16

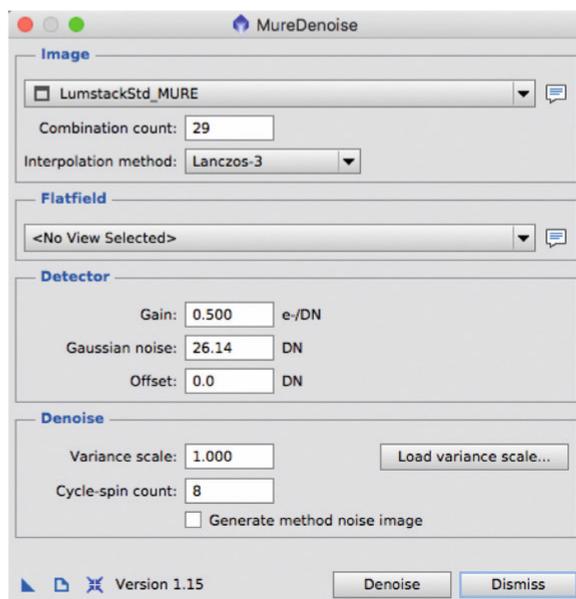


fig.2 Under the hood of this simple-looking tool is a sophisticated noise reduction algorithm that is hard to beat on linear images. Its few settings are well documented within the tool. It is easy to use too.

DN. If your image has significant vignetting or fall-off, the shot noise level changes over the image and there is an option to include a flat frame reference. The script is well documented and provides just two adjustments; Variance scale and Cycle-spin count. The former changes the aggression of the noise reduction and the latter sets a trade-off between quality and processing time. The Variance scale is nominally 1, with smaller values reducing the aggression. Its value (and the combination count) can also be loaded from the information provided by the ImageIntegration tool; simply cut and paste the Process Console output from the image integration routine into a standard text file and load it with the Load variance scale button. In practice, the transformation is remarkable and preferred against the other noise reduction tools (fig.4), providing it is used as intended, on linear images. It works best when applied to an image stack, rather than separate images which are subsequently stacked. It also assumes the images in the stack have similar exposures.

Multiscale Transforms

MLT and MMT are two multiscale tools that can sharpen and soften content at a specific image scale. We first consider their noise reduction properties and return to them later for their sharpening prowess. Both work with linear and non-linear image data and are most effective when applied through a linear mask that protects the brighter areas with a higher SNR. Unlike MureDenoise, they work on the image data rather than using an estimate of sensor characteristics. In both cases the normal approach is to reduce noise over the first 3 or 5 image scales. Typically there is less noise at larger scales and it is normal to decreasingly apply their noise reduction parameters at larger scales. Both MLT and MMT are able to reduce noise and sharpen at the same time. I used this approach for some time and for several case studies. After further research, I discovered this approach is only recommended with high SNR images (normal photographic images) and is not optimum for astrophotography. It is better to apply noise reduction and sharpening as distinct process steps, at the optimum point in the workflow. In the same manner, both tools have real-time previews and the trick is to examine the results at different noise reduction settings, one scale at a time. I also examine and compare the noise level at each scale using the ExtractWaveletLayers Script. It is a good way to check before and after results too. In doing so one can detect any trade-off, at any particular scale, between noise reduction and unwanted side-effects.

The two tools work in a complementary fashion and although there have many similarities, it is worth noting their differences:

MultiscaleLinearTransform

MLT works well with linear images and with its robust settings can achieve a smooth reduction in either color or luminance noise. It is more effective than MMT at reducing heavy noise. If it is overdone it can blur edges, especially if used without a mask, and at the same time aggressive use may create single black pixels too. It has a number of controls that provide considerable control over the outcome. At the top is the often overlooked Algorithm selection. The Starlet and Multiscale linear algorithms are different forms of multiscale analysis and are optimized for isolating and detecting structures. Both are isotropic,

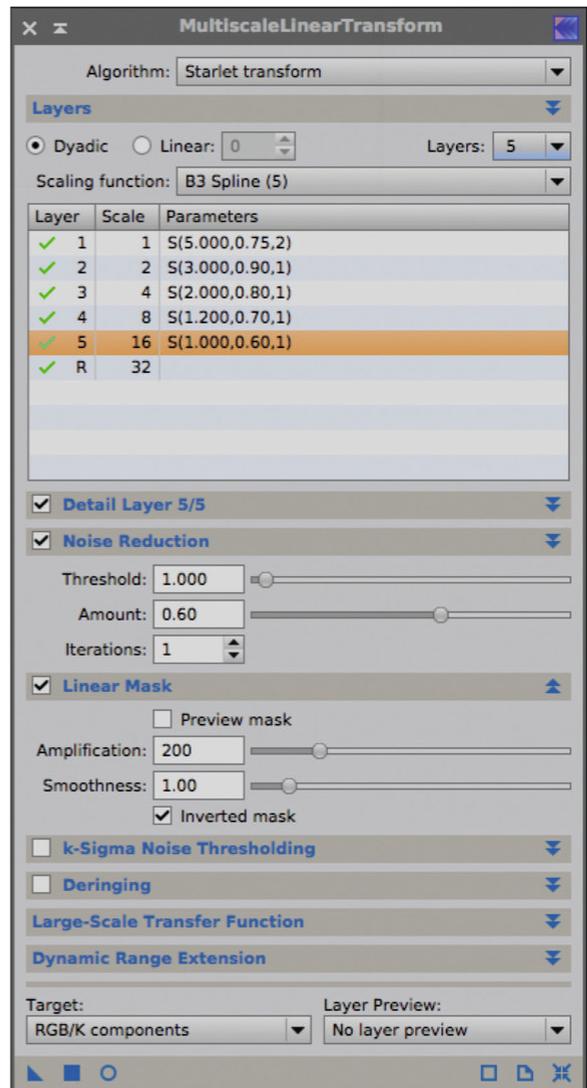


fig.3 Some typical noise reduction settings for MLT when operating on a linear image. This tool comes close to the performance of the MureDenoise script but has the advantage of being more selective and tunable.

in that they modify the image exactly the same way in all directions, perfect for astrophotography. The differences are subtle; in most cases, the scales form a geometric (dyadic) sequence (1, 2, 4, 8, 16 etc.) but it is also possible to have an arbitrary set of image scales (linear). In the latter case, use the multiscale linear algorithm for greater control at large scales. I use the Starlet algorithm for noise reduction on linear images.

The degree of noise reduction is set by three parameters: Threshold, Amount and Iterations. The Threshold is in Mean Absolute Deviation units (MAD), with larger values being more aggressive. Values of 3–5 for the first scale are not uncommon, with something like a 30–50% reduction at each successive scale. An Amount value of 1 removes all the noise at that scale and again, is reduced for larger scales. Uniquely, the MLT tool also has an Iterations control. In some cases, applying several iterations with a small Amount value is more effective and worth evaluating. The last control is the Linear Mask feature. This is similar to the RangeSelection tool, with a preview and an invert option. In practice, with Real-Time Preview on, check the Preview mask box and Inverted mask box. Now increase the Amplification value (around 200) to create a black mask over the bright areas and soften the edges with the Smoothness setting. The four tool sections that follow (k-Sigma Noise Thresholding, Deringing,

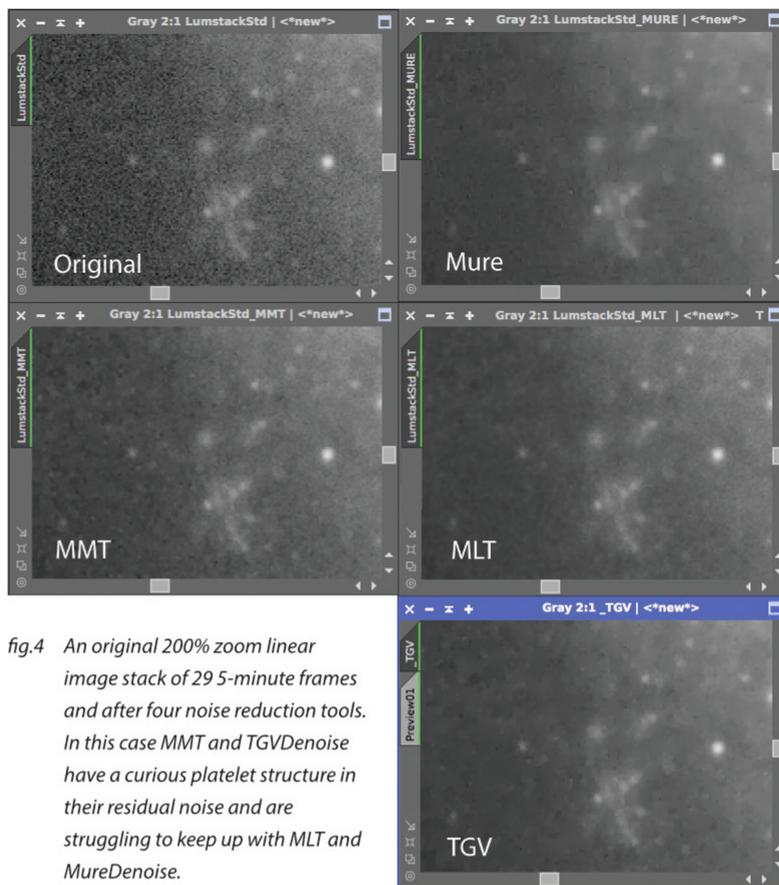


fig.4 An original 200% zoom linear image stack of 29 5-minute frames and after four noise reduction tools. In this case MMT and TGVDenise have a curious platelet structure in their residual noise and are struggling to keep up with MLT and MureDenoise.

Large-Scale Transfer Function and Dynamic Range Extension) are not required for noise reduction and left disabled. The combinations are endless and a set of typical settings for a linear image is shown in fig.3. With the mask just so, clear the Preview mask box and either apply to a preview or the entire image to assess the effect.

noise reduction tool	linear	non-linear	small scale	medium scale	retain detail	chroma noise	artifacts	comment
MUREDenoise	+++	+	+++	++	+++	mono only	+++	use before any sharpening, best for linear images, easy to adjust
MLT	++	+++	+++	++	+	option	+	use with linear mask, use starlet transform and tune with amount
MMT	+	+++	++	+++	++	option	++	use with linear mask and median wavelet transformation
TGVDenise	++	++	++	++	+++	option	+	use statistics to set edge protection level, very sensitive!

fig.5 This table is a broad and generalized assessment of the popular noise reduction techniques that looks at what they are best applied to, whether they work well with small and medium scale noise, retain essential detail, color options and potential side-effects.

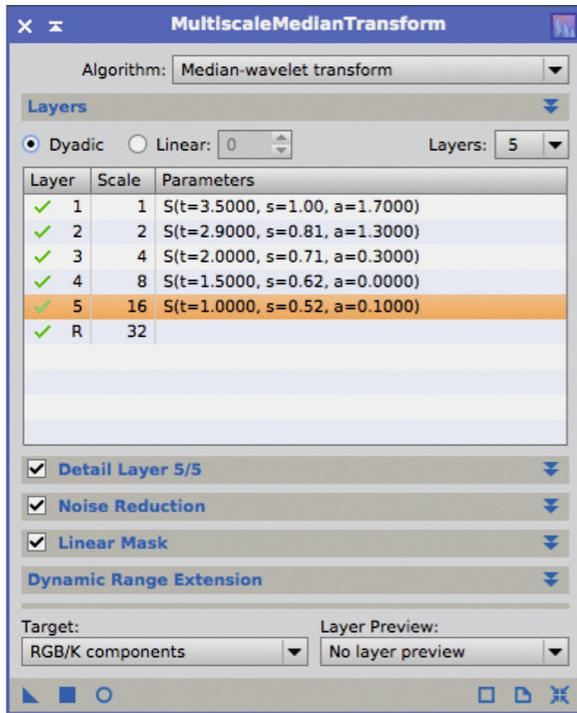


fig.6 These MMT settings were used to produce its noise reduction comparison in fig.7. Note an Adaptive setting >1.0 is required at several scales, to (just) remove the black pixels that appear in the real-time preview.

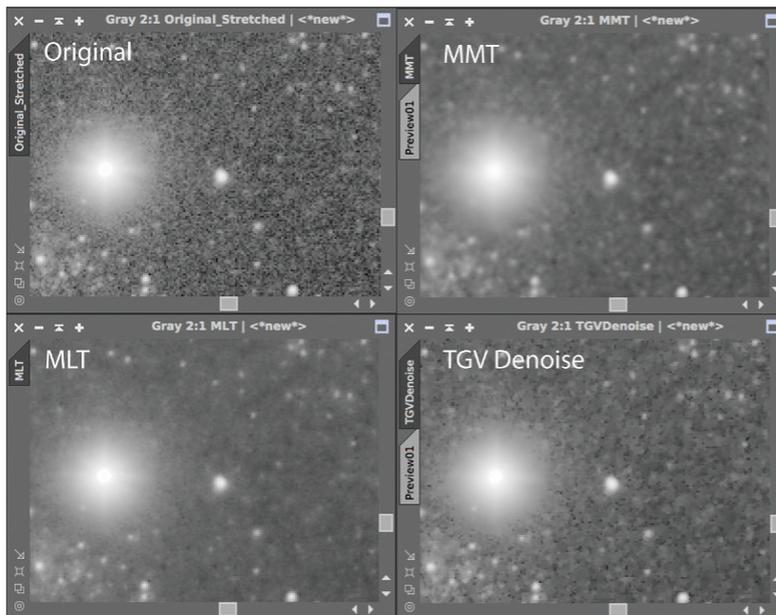


fig.7 A 200% zoom comparison of the three noise reduction tools on a noisy non-linear image after an hour of experimentation. MLT performed well, followed by MMT and TGV Denoise. In brighter parts of the image, MMT edges ahead. TGV Denoise preserves edges better, but it is easy to overdo.

MultiscaleMedianTransformation

MMT works with linear and non-linear images. It is less aggressive than MLT and a setting gives broadly reproducible results across images. It delivers a smooth result but has a tendency to leave behind black pixels. Like MLT, it is best used in conjunction with a Linear mask. It is more at home with non-linear images and its structure detection algorithms are more effective than MLT and protect those areas from softening. In particular, the Median Wavelet Algorithm adapts to the image structures and directs the noise reduction where it is most needed. The noise controls look familiar but the Iteration setting is replaced by an Adaptive setting (fig.6). Adjust this setting to remove black pixel artefacts. In this tool, the degree of noise reduction is mostly controlled by Amount and Threshold. It is typically less aggressive than MLT and can withstand higher Threshold values. In fig.4, which assesses the four tools on a linear image, it struggles. The second comparison in fig.7, on a stretched image, puts it in a much better light.

TGV Denoise

This tool uses another form of algorithm to detect and reduce noise. The settings for linear and non-linear images are very different and it is tricky to get right. Unlike the prior three tools, this one can simultaneously work on luminance and chroma noise with different settings. The most critical is the Edge protection setting, get it wrong and the tool appears broken. Fortunately, its value can be set by image data statistics: Run the Statistics tool on a preview of blank sky. In the options, enable Standard Deviation and set the readings to Normalized Real [0,1]. Transfer this value to the Edge protection setting and experiment with 250-500 iterations and the strength value. If small structures are becoming affected, reduce the Smoothness setting slightly from its default value of 2.0.

Just as with MMT and MLT, this is best applied selectively to an image. Here the mask is labeled Local support, in much the same way as that used in deconvolution. It is especially useful with linear images and reducing the noise in RGB images. The local support image can be tuned with the three sliders (histogram sliders) that change

the endpoint and midpoint values. The default number of iterations is 100. It is worth experimenting with higher values and enabling Automatic convergence, set in the region 0.002–0.005. If TGVDenoise is set to CIE L*a*b* mode, the Chrominance tab becomes active, allowing a unique setting to reduce chrominance noise. Though tricky to master, TGVDenoise potentially produces the smoothest results, think Botox.

Sharpening and Increasing Detail

If noise reduction was all about lowering one's awareness to unwelcome detail, sharpening is all about drawing attention to it, in essence by increasing its contrast. In most cases this happens with a selective non-linear transform of some kind. As such, there is an inevitable overlap with general non-linear stretching transformations. To make the distinction, we consider deconvolution and small-scale feature/edge enhancement as sharpening actions (using MLT, MMT and HDRMultiscaleTransform). In doing so, we are principally concerned with enhancing star appearance and the details of galaxies and nebulae. Deconvolution and star appearance are special cases covered in their own chapter, which leaves small-scale / edge enhancement. (Masked Stretch and LocalHistogramEqualization equally increase local contrast, but typically at a large scale and are covered in the chapter on image stretching.) Sharpening is more effective and considerably more controllable on a stretched non-linear image. Yes, it can be applied to linear images but if you consider the stretching process to follow, the slightest issue is magnified into something unmanageable.

Beyond UnsharpMask

When looking at the tools at our disposal, Photoshop users will immediately notice that one is noticeable by its absence from most processing workflows. UnsharpMask is included in PI but it is rarely used in preference to deconvolution and the multiscale tools. It creates the illusion of sharpness by deliberately creating alternating dark and light rings around feature boundaries and in doing so destroys image information. Deconvolution, on the other hand, attempts to recover data. UnsharpMask does have its uses, mostly as a final step prior to publication to gently add sparkle to an image. The trick is to view the preview at the right size to assess the impact on the final print or screen.

The multiscale tools include our old friends MLT and MMT but here we use them in a different way to give the appearance of sharpening by changing local contrast at a particular scale. As before, they are best used selectively through a mask. This time, however, the

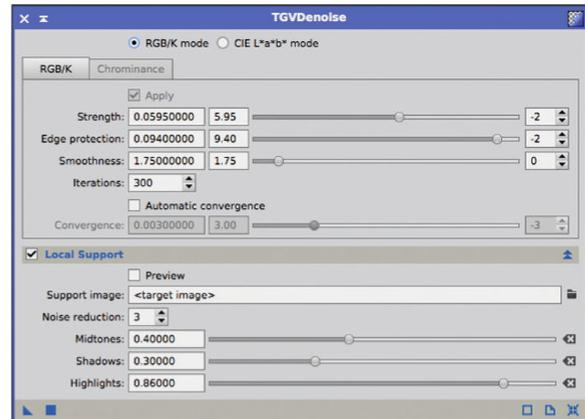


fig.8 These settings were used to produce the noise reduction comparison in fig.7. A small increase in the strength caused unsightly platelets to appear in the background.

linear mask is non-inverted, to protect background areas of low SNR. To sharpen an image, the Bias setting in either tool is increased for a particular layer, corresponding to a different image scale. Visualizing scale is daunting for the novice and it is useful to run the ExtractWaveletLayers image analysis script on the target image. The multiple previews, each extracting the information at a different image scale, provide a useful insight into the image detail at each.

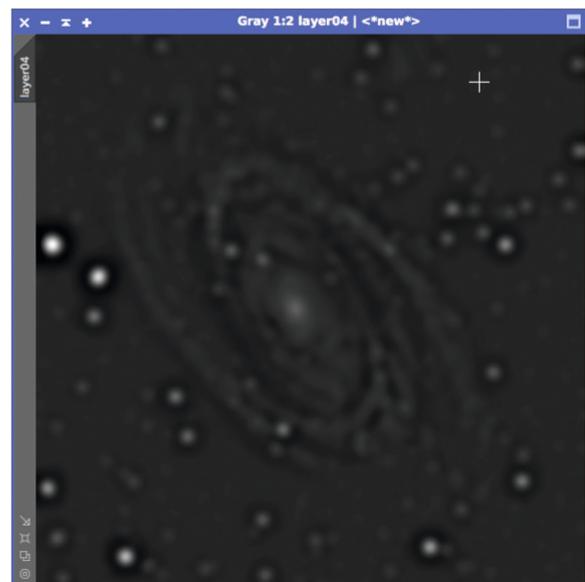


fig.9 This image was generated by the ExtractWaveletLayers script. This one is for scale 5 and shows the broad swirls of the galaxy arms. Increasing the bias of this layer, increases its contrast and its emphasis in the combination of all the scales and the residual layer.

These images are typically a mid-grey, with faint detail etched in dark and light grey (fig.9). From these, one can determine where the detail and noise lay and target these layers with noise reduction and sharpening. These images also give a clue on how the tool sharpens and the likely appearance: When these images are combined with equal weighting, they recreate the normal image. The bias control amplifies the contrast at a particular scale so that when it is combined, the local contrast for that scale is emphasized in the final image. For that reason, it is easy to see that the general tone of the image is retained but the tonal extremes are broadened.

As an aside, the same bias control can be used to de-emphasize structures too, by reducing its value to a negative number. It is not unusual to see the first layer (corresponding to a scale of 1) completely disabled in an RGB image to remove chroma noise. It has other potential uses too: A globular cluster technically has no large-scale structures but some of the brighter stars will bloat with image stretching. Applying MLT (or MMT) with the bias level slightly reduced for layers 4–5, reduces the halo around the largest stars.

The other common characteristic of these two sharpening algorithms is their tendency to clip highlights. This is inevitable since sharpening increases contrast. Both multiscale tools have a Dynamic Range Extension option that facilitates more headroom for the tonal extremes. I start with a High range value of 0.1 and tune so the brightest highlights are in the range 0.9–0.95. Both tools have a real-time preview facility and, in conjunction with a representative sample preview, enable almost instantaneous evaluation of a setting.

Sharpening with MLT

MLT can be used on linear and non-linear images. In common with all other linear sharpening tools, it can produce ringing around stars, especially on linear images. For that reason, it has a deringing algorithm option to improve the appearance. In a number of tutorials MLT, is commonly used with its linear mask to exclude stars and background, with a little noise reduction on the first layer and a small bias increase at the larger scales to make these more pronounced. Compared with MMT, MLT works best at medium and large scales. As usual, ensure one has a screen transfer applied to the image before activating the real-time preview, to assess the likely impact on the final image. Some typical settings in fig.10 were applied to a stretched image of M81 (fig.14). The MLT tool did the best job of showing the delicate larger structures in the outer galaxy arms. It was less suited for enhancing fine detail.

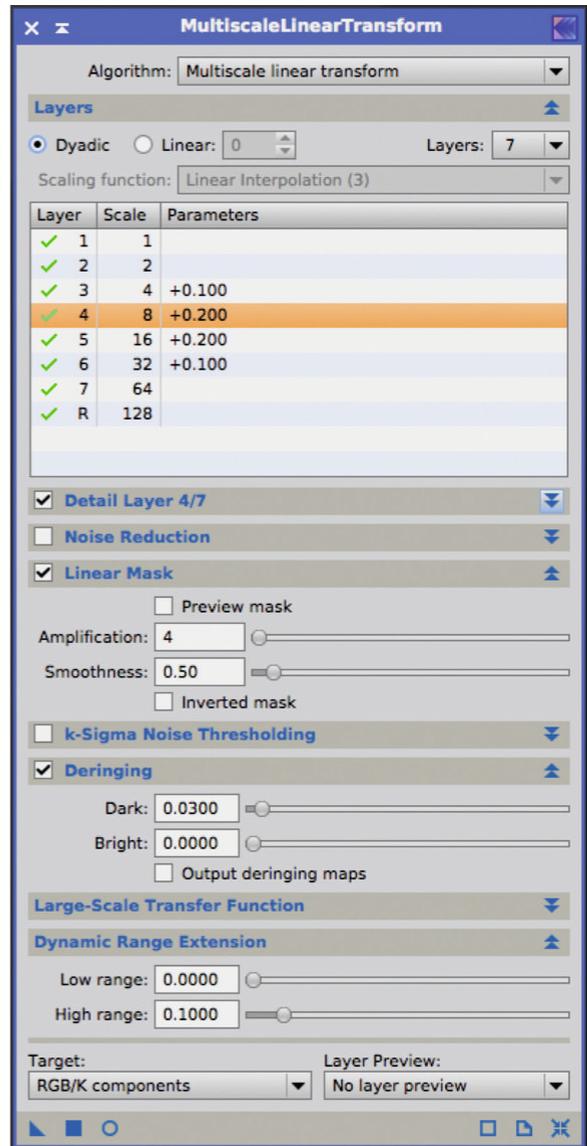


fig.10 These settings were used to produce the sharpening comparison in fig.14. Note the bias settings are working on the larger scales and a small amount of deringing keeps artefacts in check. Sharpening increases dynamic range and here it is extended by 10% to avoid clipping.

Sharpening with MMT

MMT improves upon MLT in a number of ways. MMT does not create rings and sharpens well at smaller scales. The multiscale median algorithm is not as effective at larger scales, though the median-wavelet transform algorithm setting blends the linear and median algorithms for general use at all scales.

MMT can produce other artefacts, which is usually an indication of over-application. Again, it is best to

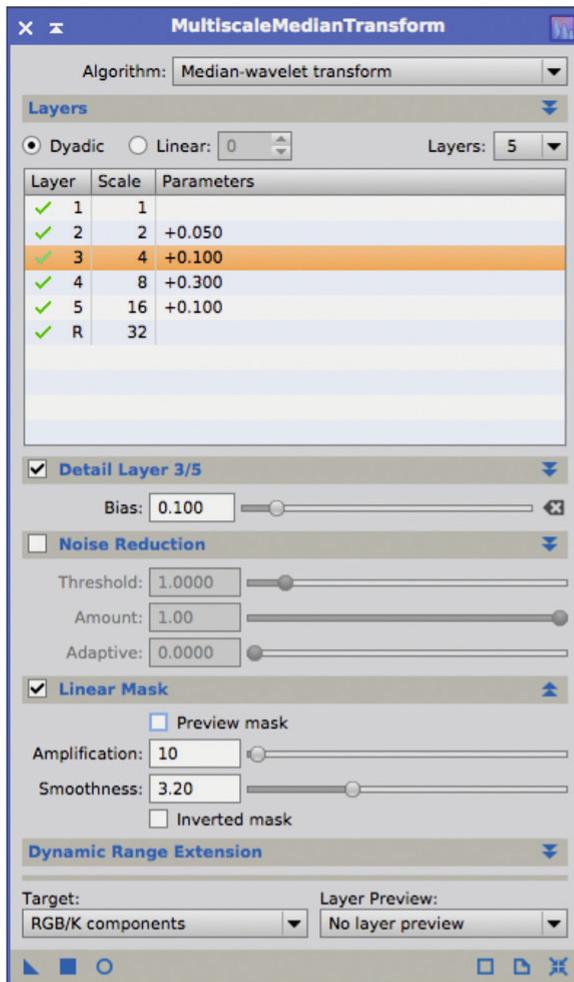


fig.11 These settings were used to produce the noise reduction comparison in fig.14. The bias settings here have an emphasis on smaller scales and deringing is not an option or required. Sharpening increases dynamic range and here it is extended by 10% to avoid clipping.

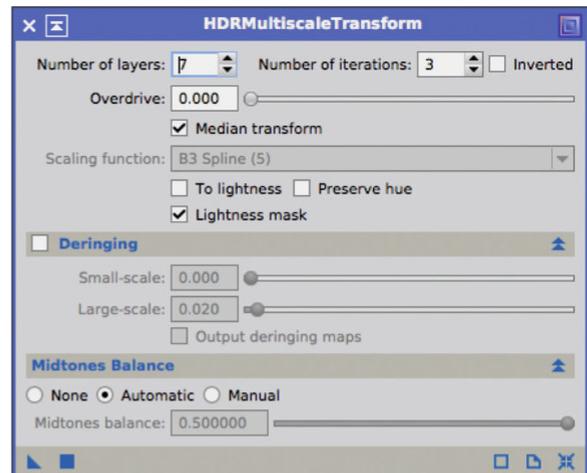


fig.12 Likewise, the simpler settings for HDRMT create a wealth of detail where there apparently is none. The scale factor changes what is emphasized, from subtle swathes in brightness to highlight local changes from dust lanes.

examine the extracted layer information to decide what to sharpen and by how much.

Sharpening with HDRMT

This tool is simpler to operate than the other two multiscale tools. It works in a very different way and as the name implies, it is used with images of high dynamic range. It has the ability to create spectacular detail from a seemingly bright, diffuse galaxy core. I usually apply this selectively to a non-linear image to enhance nebula or galaxy detail, using the median transform option. Changing the layer value generates diverse alternatives. One does not have to choose between them, however, simply combine these with PixelMath. In common with other tools, it has an in-built lightness mask and deringing option, if required. The Overdrive setting changes the amount of tonal

sharpening tool	linear	non linear	small scale	medium scale	ringing	artifacts	local support	comment
HDRMT	+	+++	+++	+++	++	+	+++	use to enhance bright structures
MLT	+	++	++	+++	+	+	+++	use the starlet algorithm for medium scales
MMT	++	+++	+++	++	+++	++	+++	use median wavelet algorithm for best results over range of scales

fig.13 As fig.5, but this time a comparison of sharpening tools. This is generalized assessment of the performance of these tools on linear and non-linear images, optimum scale and likely side-effects, based on personal experience and the tutorials from the PI development team. At the end of the day, further experimentation is the key to successful deployment.

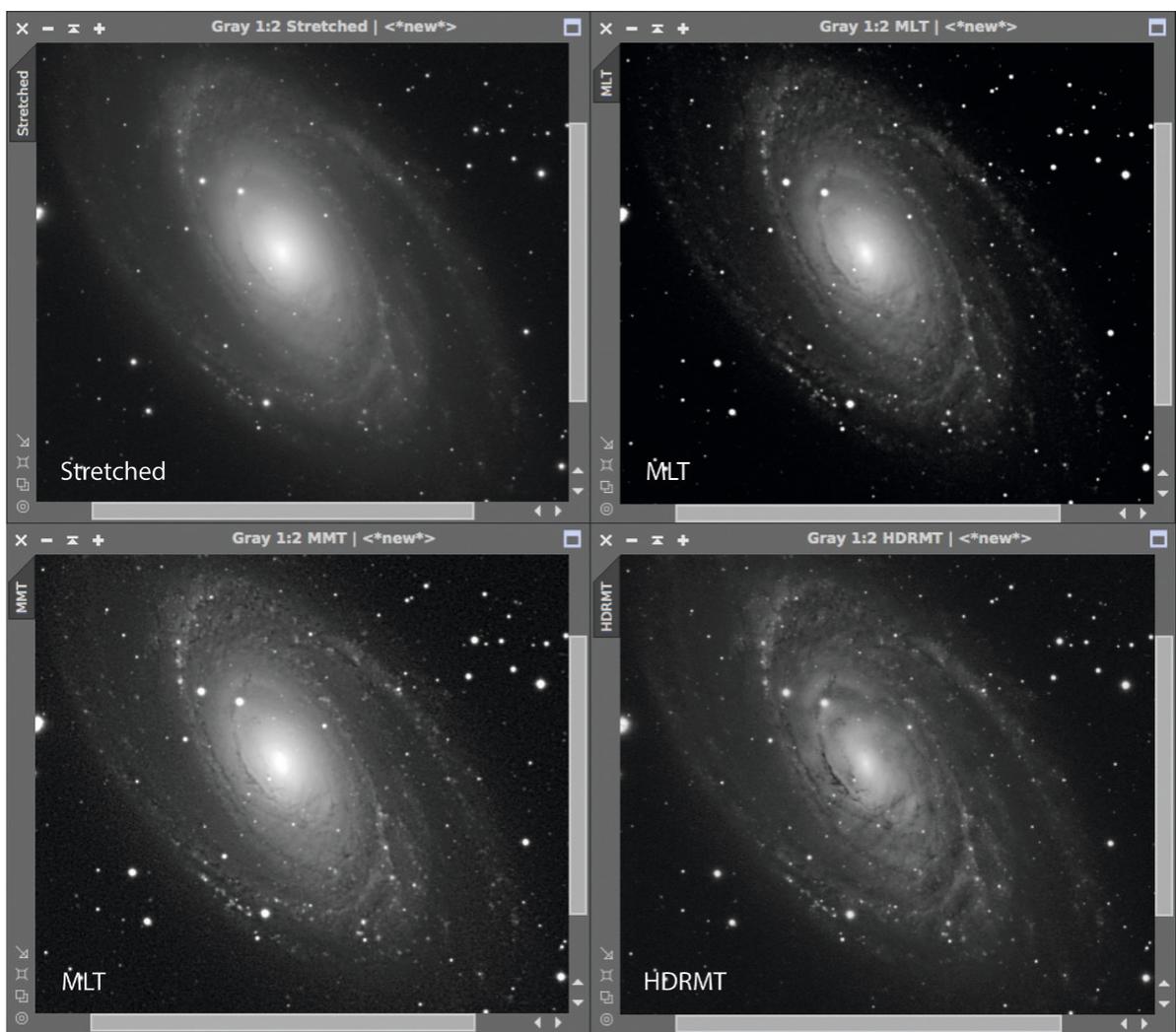


fig.14 A comparison of sharpening techniques on the delicate spirals of M81 (shown at a 50% zoom level to see the effect on the larger scales at which sharpening operates). These are not the last word in sharpening but give an appreciation of the very different results that each of the tools can bring. MLT and MMT are subtly different in output, with MMT being more adaptable. HDRMT is particularly dynamic. The result here is quite tame compared to some that can occur. The trick is to realize that HDRMT is not a silver bullet, just a step on the journey and the result can be subsequently stretched or blended to balance the galaxy's overall brilliance with the surrounding sky.

compression and, with the iterations setting, provides an opportunity for fine-tuning.

Combining Strengths

Some of these tools are quite aggressive or dramatically change the image balance. Subsequent processing, for example, Curves Transformation, can recover this. Another possibility is to create a number of sharpened versions, optimize for different effect and then blend them. The most convenient way to do this is to simply add them using a simple PixelMath equation, which additionally provides

endless possibilities to weight their contribution in the final image. For example, one of the drawbacks of sharpening tools is their clipping effect on stars. Even with additional headroom, stars become too dominant. One method is to apply the HDRMT tool to a stretched image and then blend this image with another optimized for star processing, with a similar median background level. For example, apply the MaskedStretch tool to a linear version of the same file for star appearance and blend the large-scale features created by the HDRMT tool with the small-scale structures from the MaskedStretch version.