STEP 1. Sorting out our images

The common technique of capturing many exposures of the same target with the same settings still applies with DSLRs. Even more so, since averaging them out later will significantly reduce the noise in the final image. This is because your target will always fall in the same place but the random noise produced by the camera will be just that, random. When you average out a number of images, effectively the noise is cut down but your target is maintained. This improves the Signal-to-Noise Ratio (SNR) of your image. Since DSLRs are generally uncooled (unless modified) and have a lot more electronics than astronomical CCD cameras, this technique of averaging multiple exposures of the same thing is particularly important. We note immediately that exposures with the DSLR should be captured in **RAW**, not *JPEG*. Astrophotography tends to require extensive post-processing for ultimate results and *8-bit JPEGs* are extremely counter-productive due to the compression losses.

To further aid in cleaning up the final image, it is a good idea to capture *dark* and *bias* calibration images. The dark images are captured by placing the cap on the camera (or the lens) and at more or less the same temperature and the exact same exposure settings (length, ISO, etc), capturing images with no external light falling on the sensor. The resulting image will be one of just noise and hot pixels. The bias images are captured by doing the same but setting the exposure time to the minimum possible on the DSLR, as this is only intended to capture the signal produced by physically reading the sensor. Furthermore, to remove vignetting inherent to the optical system, *flat* calibration images can be captured. These simply involve using the exact same equipment used for imaging, shooting a white, evenly illuminated surface (the result is simply the vignetting pattern). Capturing a number of these calibration images (5 or 10) is a good idea to average them individually (to produce a master dark, master bias and master flat). These are then used to *calibrate* the images of your target, while averaging them out. This tutorial discusses calibration and images used in detail and this tutorial discusses how to take your target, dark, bias and flat images and produce one final, calibrated result.

Once processed, you should have one final image of your target, calibrated and ready for post-processing. It is important that this final image be in **16-bit**, and **TIFF** is a good format (though **FITS** is even better). An *8-bit JPEG* will be extremely, extremely difficult to post-process later due to the inherent issues presented by the image compression.

STEP 2. Minor cropping out of sensor defects

It is fairly common for the image to have darkened borders. To check, simply perform a quick auto-stretch of the image by clicking the **STF AutoStretch** button on the top-right of PixInsight.

The light pollution captured may be evident at this stage. If flats images were not used in calibration, you will also observe the vignetting pattern (see above). On zooming in on the top border, one can see the darkened band.

This is visible to all four sides of the images and should be cropped out. We do this with the **DynamicCrop** tool. Open this tool and click its **Reset** button. Now we can drag slightly inward from all four sides of the image, to the point of excluding these bands along the sides. Once done, click the **Execute** button on the tool to crop out these darkened bands.

Close **DynamicCrop** and now save the image. It would be a good idea at this point to save the image in a more familiar format (for astrophotography post-processing), **FITS**. Therefore, save the image as a new file and select **FITS** for format. Enter a new filename such as the original one with *_Crop* added on to the end. When asked for bit rate, select **64-bit IEEE 754 floating point**. This will allow us to maintain the dynamic range of the data intact while we further post-process. Though 64-bit may be overkill (32-bit should suffice), it is not going to pose a problem to work with. Once the image is saved, close it and re-open it.

STEP 3. Removing the background gradient

Whether or not light pollution is as prominent in your image as in this one we are dealing with here, it is always a good idea to go through this step in order to remove any background gradients in the image and leave behind a very clean image. The powerful **DynamicBackgroundExtraction** tool comes into play here. With your image auto-stretched already, open this tool. Once open, click the **Reset** button on it to initialise it on your image.

This tool will allow us to pick, or automatically place, sampling points all over the image. The tool then uses these sampling points to build an image of what should be background, later subtracting this from the original image. This will effectively remove the gradient in the background and also null light pollution. To start, we expand the **Sample Generation** and **Target Image Correction** tabs.

We are going to simply allow the tool to automatically place the sampling points all over the image. Change the **Default sample radius** to **15** and **Samples per row** to **25** (to add more sampling points, of larger radius). Also alter **Minimum sample weight** to **0.100** to allow for more samples to be used. Click the **Generate** button and samples will appear all over the image.

Before we continue, we must first see what we have. Above, some samples appear white and others appear red. The white ones are included by the tolerance value set in the tool. The red ones were placed but are not included due to the tolerance value being too low. What is important is that these samples must not lie on top of any stars or any nebulosity we care about, just background. For now let us leave them where they are. We should encompass the entire image with samples and this will require us to increase the tolerance value. The default value of **0.500**. I increased mine to **1.000** and clicked **Generate** again to check the resulting samples.

We are getting closer, but the two top corners are still excluded and there are samples here that are still being rejected (red squares). It is important that the tool is able to use all four corners and sides of the image to produce a good background gradient model. We increase tolerance slightly to **1.250** and click **Generate** again to check.

Looking much better now, with the entire image covered and all samples accepted. In practice, tolerance should be set to the lowest value possible as long as all our desired samples are included. **1.250** may work but perhaps slightly lower also works. We lower the value to **1.200** and click the **Resize All** button to check to see if any samples appear rejected (red squares). None do but **1.150** caused one sample to be rejected so **1.200** is the value to use.

Now that the entire image is covered in samples and all are accepted by tolerance, we should go through all the samples to make sure they do not overlap any stars or nebulosity. If they do, they can either be moved around or selected (by clicking) and deleted individually. It helps to zoom in and scroll across and down the image to check samples closely.

Above shows some samples moved, others removed entirely. Samples can also be added by clicking around the image. Once the samples are ready, select **Subtraction** from **Correction** and check **Discard background model** (if you want to see the actual image built from this model and extracted, leave this unchecked). Since we may use this tool again on the same image, we should save its state for later use. In fact, we may apply this process twice to really clean up the image. To save its state, drag the button on the lower-left corner of the tool, called **New Instance**, to anywhere on the PixInsight grey area (workspace).

This will create an icon, by default called *Process01*. It can be renamed to whatever you want by right-clicking it and going to **Set Icon Identifier**. I renamed mine to *DBE*, for clarity. Once the tool's state is saved, click its **Execute** button to perform the background modelling and extraction of your image. A new image will appear, click the **STF Auto-Stretch** button in PixInsight (top-right) to check it out.

This is looking infinitely better already, but it may be worth running the tool a second time to clean up the image further. To do this, close the tool and the previous image, only leaving the new image open. Re-open **DynamicBackgroundExtraction** on the new image and click its **Reset** button to initialise it.

Since we saved the tool's state from earlier, we can re-use it. Drag and drop our created icon from the PixInsight workspace to the bottom bar of **DynamicBackgroundExtraction** - the bar where the bottom six buttons are.

You will quickly notice all your hard work of moving samples re-appear all over your image - excellent! Now let us lower the tolerance value until samples start to be rejected. Set **Tolerance** back down to the default **0.500** value and click the **Resize All** button. It is important not to click the **Generate** button because this will re-do the placement of the samples. **Resize All** will just alter the samples already placed with the new values you enter.

Since all the samples seem to still be accepted, we leave tolerance alone as **0.500** is a good value. Feel free to move some samples around a bit if you notice they are overlapping new-found nebulosity (that you did not see earlier due to light pollution or other sources of background gradient). We note the large lightened band along the bottom of the image. There are no samples along this and it may be a good idea to have some there to account for it in the background extraction. I just clicked along it to add samples to include it. In order to remove the background a little bit more precisely, it can be made slightly less smooth and more exact to the sample placement. To do this, lower the **Smoothing factor**. I lowered mine to **0.100** to get a more exact extraction. Once happy, click the **Execute** button again and you will again have a new image. Click **STF Auto-Stretch** to check it out.

The brightened band along the bottom is still present but a bit more tamed. These tend to appear in DSLR images because of the internal electronics. Since the target does not lie along the band, I can remove this band entirely by cropping out this lower part of the image using **DynamicCrop**.

This image should be saved as a new image. I called mine *RGB* as this will be the colour image I deal with now. I saved mine in **64-bit IEEE 754 floating point**. Close the image once saved and re-open it.

STEP 4. Extracting a Luminance image from the RGB image

A good technique in post-processing astronomical images is to use a Luminance image (a monochrome image of the entire visible spectrum together) to bring out the detail in a heavily-processed RGB colour image that has received a lot of treatment to noise. I tend to go as far as blurring the RGB image entirely to maintain only colour information, to later combine with the Luminance image to bring out the detail. We note that because the camera used is a One Shot Colour camera (e.g. a DSLR), this is not a proper Luminance image but it works as a technique for post-processing. Since the DSLR has captured the image *as is*, we need to extract a Luminance image from what we already have. This is done after elimination of background gradients and light pollution. Before we do this however, we need to make sure that all three colours, Red, Green and Blue are weighted the same for this extraction. With the colour image opened, open the **RGBWorkingSpace** tool and expand the **Luminance Coefficients (D50)** tab.

To correct the weighting, set **Red**, **Green** and **Blue** to **1.000000** and click the square **Apply** button.

With the RGB working space now set correctly, close **RGBWorkingSpace** and open the **ChannelExtraction** tool and click the **CIE L*a*b*** radio button.
Here, we uncheck **a** and **b** and only leave **L** checked. Click the square **Apply** button to extract the Luminance image.

The tool can now be closed and the Luminance image can be inspected. Click the **STF Auto-Stretch** button to check it out.

This is a Luminance image we will use later in post-processing. Save the image with a filename *L*, also in **64-bit IEEE 754 floating point**. Now both images are ready for individual post-processing.

STEP 5. Post-processing of the RGB colour image

Open the *RGB* image we saved earlier after extracting the background gradients. The first step to carry out in the post-processing of this image is colour calibration. This ensures Red, Green and Blue lie in the same place and therefore reproduces colour accurately. We will then remove a lot of the small scale noise in the image. After, we will stretch the histogram to strongly enhance image contrast and finish with blurring the image entirely to only retain colour information.

First click the **STF Auto-Stretch** button to see what we are working with. To perform a colour calibration, we need to provide PixInsight with an area of the image that is purely background and another area of the image that is something we are interested in - nebulosity. To best do this, we will create two *Preview Boxes*, sectioned areas of the image that are used by tools. Click the **New Preview Mode** button along the top of PixInsight.

Drag and create a *Preview Box* in a large area that is basically just background (space). Stars are alright to be included here, but avoid all nebulosity. I created a large one on the top-left corner of the image.

To not confuse things, go to **Preview** and click **Modify Preview**. Enter a new name, such as *Background*, and click **OK**. Now create another *Preview Box* but over a group of stars or even a galaxy (if your image contains this). The idea is that the tool will use the bright, white objects as a reference for what should be white. You may need to click the **New Preview Mode** button again. Once done, rename this preview box like you did the other. I renamed mine to *White*.

Click the **Readout Mode** button on the top of PixInsight (near the **New Preview Mode** button) to stop yourself accidentally creating new preview boxes or modifying existing ones. We are now ready to perform a colour calibration. First, we will neutralise the background with **BackgroundNeutralization** so open this tool.

Here, we must select the *Background* preview box from the top list. Once done, simply click the square **Apply** button.

You may notice a slight change to your image - the background should take a more neutral tone. Close the tool and now open the **ColorCalibration** tool.

Here we do a similar thing. From the list on the top, under **White Reference**, select the *White* preview box. From the list on the bottom, under **Black Reference**, select the *Background* preview box. Once done, leave everything else at default and click the square **Apply** button.

Once done, close this tool. We can now delete the preview boxes. To do so, go to **Preview** and click **Delete All**. You may need to click the **STF Auto-Stretch** button again to see the result of the colour calibration process. You may notice a slight green tint to your image. This is normal because the sensor is more sensitive to green than the other parts of the spectrum. Thankfully PixInsight has an excellent tool to deal with this. Run the **SCNR** tool and on default settings, just click the square **Apply** button on it.

Again, you may need to click the **STF Auto-Stretch** button again to see the result of the colour calibration process. We are now done with the colour calibration process and you should save the image as a new one. I added *_ColourCal* to the filename.

Before we stretch the histogram to reveal all the wonderful contrast in the image, we should take advantage of the image's current linear state (unstretched historgram) to remove small scale noise. To do this, we should apply a mask to the image so we remove noise from where it is most present - areas of weak signal (the background). To create the mask, first duplicate the image. Right-click on it and click **Duplicate**. With the original image selected, make sure it is not being automatically stretched (click **Reset Screen Transfer Function** on the top-right of PixInsight). Open the **ScreenTransferFunction** tool. In this tool, click the **Auto Stretch** button.

Now open the **HistogramTransformation** tool.

The **ScreenTransferFunction's Auto Stretch** is not doing anything to the image. It is simply altering the way we are seeing the image. We should however transfer this to the **HistogramTransformation** tool so that this transformation is applied to the duplicate image. To do this, click to select the original image (the one with the **Auto Stretch**), then drag and drop the **New Instance** button from **ScreenTransferFunction** to to the bottom bar of **HistogramTransformation**.

The straight line on the **HistogramTransformation** line should turn into a sharp curve. This is the function that needs to be applied to the duplicate image. To do this, click to select the duplicate image and click the square **Apply** button on **HistogramTransformation**.

The original image is still linear - nothing has been stretched, even if it looks like it has (**Auto Stretch** is just for the user to see what is there). The duplicate image however is stretched and can now be used as a mask *as is*. Close the **HistogramTransformation** tool and then click to select the original image. Go to **Mask** and click **Select Mask** (or press **CTRL M**). From the list, select the duplicate image and click **OK**.

The mask is now applied to the original image. We note that the red areas of the image are *protected* and will not receive treatment. However, we need to *attack* the background areas so we need to invert the mask to instead protect the areas of strong signal (the nebulosity). To do this, go to **Mask** and click **Invert Mask** (or press **CTRL SHIFT I**).

We are now ready to start removing small scale noise. The best tool to start with is **ATrousWaveletTransform**, so open this tool.

The good thing about using this tool is that it can apply noise reduction to various layers of pixel scale. Since we are dealing with small scale noise, we will mainly target the first layer. We will also target higher layers but in decreasing strength. Check to enable **Noise Reduction** for layers **1** to **4**. For layer **1**, set **Threshold** to **4.000** and **Iterations** to **3**. For layer **2**, set **Threshold** to **2.000** and **Iterations** to **2**. For layer **3**, set **Threshold** to **1.000** and **Iterations** to **2**. For layer **4**, set **Threshold** to **0.500** and **Iterations** to **1**. Once done, click the square **Apply** button.

The noise has clearly improved considerably. Now close **ATrousWaveletTransform** and open the next tool, **MultiscaleMedianTransform**.

In this tool, we replicate the settings. Check to enable **Noise Reduction** on layers **1** to **4**. Set **Threshold** to **4.000**, **2.000**, **1.000** and **0.500** respectively. Click the square **Apply** button once done.

A minor but welcome improvement to noise. We can now close **MultiscaleMedianTransform**. The mask can also be removed. To do this, go to **Mask** and click **Remove Mask**. The duplicate image we created earlier to act as a mask can also be closed without saving. Save the image now. I added *_NR* to the filename.

We will now stretch the histogram to reveal all the detail. For this, make sure there is no **Auto Stretch** being applied and open the **HistogramTransformation** tool. Click the **Reset** button on the tool and select the image from the list.

Originally, the histogram is piled up on the dark side of the histogram, hence the lack of contrast shown in the image in its original state. By stretching the histogram, we spread out the detected signals over a larger range of pixel values, giving rise to massively increased contrast between dark and bright areas (revealing fine detail). This is also the reason we saved the image in 64-bit format - to have a much larger range of values we can occupy for greater precision.

To stretch the histogram, first click the **Real-Time Preview** button on the bottom-left of the tool to see what we are doing in real-time. Having selected the image from the list in the tool shows us the histogram in its current state and in its stretched state. To perform the stretch, simply move the middle pointer (mid-tones slider) to the left to curve the straight line.

We can already see a massive increase in contrast from this image to the one above it. The important thing to note is that as we move the mid-tones slider to the left, the histogram is stretched and moved across to the right. If we over-do it, the histogram will start to *white clip*. This is when data goes over the saturation point and is therefore lost. We need to avoid over-doing it as a well. The first stretch can be extremely aggressive, as shown above. To apply it, just click the square **Apply** button and then click the **Reset** button to perform a second stretch.

The reason the background is turning white is because the entire histogram is shifting to the right. In theory, the histogram should start from the very beginning. To move the histogram to the left now, we simply move the left pointer (black-point slider) to the right. You will notice the background getting darker. Be careful though. You are dealing with a DSLR image. This means there is quite a bit of underlying large scale noise to the background and if you darken it too much, it will make the large scale noise very clearly visible. I would therefore only move the black-point slider to the right a bit, as shown below.

Once you click the square **Apply** button, make sure you click **Reset** before performing another stretch. The second stretch should be much less aggressive and just enough to bring out details a bit further but to avoid making everything look saturated. Feel free to also perform a bit of further black-point clipping at the same time, as shown below.

A good guideline to stop is when the histogram is nicely stretched out and lies about a third of the way up the range. Do remember that images from DSLRs have their limits in terms of the noisy background. Do not over-do the black-point clipping or stretching as it will look worse. Close the **HistogramTransformation** tool once done. Save your image. I added *_HistT* to the filename.

We can perform a bit further noise reduction before we finish up with the RGB colour image. We will again need a mask. At this stage, a good mask is a Luminance of the image's current state. We therefore perform the same technique we did in **step 4** to extract the Luminance. We use **ChannelExtraction**, in **CIE L*a*b*** mode, unchecking **a** and **b** and clicking **Apply**.

This image can now be applied as a mask by going to **Mask** and clicking **Select Mask** (or press **CTRL M**). Select the new image from the list and click **OK**. Again, invert the mask so we are *attacking* the background. Go to **Mask** and click **Invert Mask** to do this (or press **CTRL SHIFT I**).

We now re-open the **ATrousWaveletTransform** tool and click **Reset** on it to make everything default. Since we have already applied a great deal of noise reduction, we will only apply a little now. Check to enable **Noise Reduction** only for layers **1** and **2**. Set **Threshold** to **3.000** and **Iterations** to **2** for layer **1** and set **Threshold** to **0.500** and **Iterations** to **1** for layer **2**. Click the square **Apply** button once done.

Now close **ATrousWaveletTransform**. Open **ACDNR**.

We will be a bit more aggressive about our noise reduction here. This is an excellent tool as it perform great noise reduction whether the image is linear (original unstretched histogram) or non-linear (stretched histogram). Set **StdDev** to **2.0** and **Amount** to **1.00**. Amount essentially defines how much we blend the noise-reduced image with the original image. A setting of 1.00 ensures we keep the fully noise-reduced image, not a blend with the original at all. Click the square **Apply** button once done.

A very noticeable improvement to noise level achieved. Close the **ACDNR** tool. The mask can now be removed by going to **Mask** and clicking **Remove Mask**. The Luminance image created to act as a mask can now be closed without saving.

Cooled One Shot Colour CCD camera users may not need to do this but DSLR users probably will. The following is a simple technique used to reduce the large scale, blotchy, coloured thermal noise splattered all over the background. It was presented by Alejandro Tombolini in the PixInsight forums (credit goes to him for the following). We will need to create a mask to protect the strong signal areas and for this we use the **RangeSelection** tool, so open this.

Click the **Real-Time Preview** button and the **Reset** button. What you will see is a plain white image. The aim here is to increase the **Lower Limit** slider to exclude the background but include stars and the nebulosity you care about (the strong signal areas). Set **Fuzziness** to **0.10** and **Smoothness** to **2.00** and then play around with the **Lower Limit** slider until you only include the strong signal areas. Once happy, click the square **Apply** button to create the mask image.

Close the **RangeSelection** tool and apply this mask image to the RGB colour image by going to **Mask** and clicking **Select Mask** (or press **CTRL M**), selecting the new mask image from the list and clicking **OK**. Initially the mask will be protecting the background but attacking the strong signal areas, so this must be inverted by going to **Mask** and clicking **Invert Mask** (or press **CTRL SHIFT I**).

Now that we are protecting the strong signal areas, we can attack the chrominance noise. The tool of preference is again **ATrousWaveletTransform**, so open this. Click **Reset** once opened. To work on chrominance noise, we must select **Chrominance (restore CIE Y)** from the bottom-left menu, **Target**. We want to remove the detail from large scale layers so from **Layers** on the top-right, select **7**. This will expand the list of available layers. To remove large-scale noise, just uncheck **Detail Layer** for layers **4** to **7**. Once done, click the square **Apply** button.

Once done, click **Reset** and close **ATrousWaveletTransform**. Remove the mask by going to **Mask** and clicking **Remove Mask**. You can also close and not save the mask image created earlier. An inspection of the results above shows a much more neutral background with greatly reduced coloured noise. It may be possible to further black-clip the image due to the reduced background noise. You can do this with **HistogramTransformation**, enabling **Real-Time Preview** and moving the black-point slider to the right. Like earlier, avoid darkening the background too much such that blotchy noise reappears.

Once happy with the background, click the square **Apply** button and close the **Real-Time Preview** window and the **HistogramTransformation** tool. The image can now be saved again. I added *_NR* to the filename once more.

Now that our RGB colour image is amply noise-reduced and stretched, we will completely destroy the detail in it by heavily blurring the image. This may seem counter-productive but it will provide a cleaner result at the end once we re-combine with Luminance. The tool of choice is **ATrousWaveletTransform**. Make sure you click **Reset** on it to set it back to default. To blur the image, all we need to do is disable layers in order to remove fine detail in small scales. To amply blur an image, it suffices to disable layers **1** to **4**. To do this, simply select each layer and in each case, click to uncheck **Detail Layer**. Click the square **Apply** button once done.

Now close **ATrousWaveletTransform** and save the image. I added *_Blurred* to the filename. We are now done post-processing the RGB colour image and will move on to the Luminance image.

STEP 6. Post-processing of the Luminance image

With all images closed in PixInsight, let us open up the Luminance image we created earlier from our background-subtracted RGB colour image (see **step 4**).

This image contains information related to brightness all over the visible spectrum (Red, Green and Blue). The theory is simple - we will post-process this image to the point that all the detail possible is revealed and sharpened well. This image is later combined with the blurred RGB colour image to provide an image that is both rich in colour and rich in detail.

We start post-processing the Luminance image by removing small scale noise. We would normally start by extracting the background to enhance contrast and remove gradients but this has already been done in **step 3**, prior to Luminance image extraction in **step 4**. The process is identical to how we did it with the RGB colour image. We will use **ATrousWaveletTransform** but before we do anything, we must create our mask to protect the strong signal areas (the nebulosity) and attack only the weak signal areas (the background). To create the mask, right-click on the Luminance image and click **Duplicate**. Click to select the original image and then open the **ScreenTransferFunction** and **HistogramTransformation** tools.

Firstly, we click **Auto Stretch** in **ScreenTransferFunction** to perform an automatic stretch on our original image. This will reveal all the detail automatically (by enhancing contrast) but not in a permanent way (nothing is really done to the image itself, just the way it is displayed).

What we want is for our duplicate image to have this level of stretch applied to it, permanently, to use it as a mask. As we did earlier with our RGB colour image, all we need to do is drag and drop the **New Instance** button on **ScreenTransferFunction** to the bottom bar of **HistogramTransformation**. This will essentially copy over the stretch to be applied to another image.

HistogramTransformation will immediately show the aggressive stretch as a sharp curve. To apply it to our duplicate image, simply click to select the duplicate image and click the sqaure **Apply** button on **HistogramTransformation**. You can then close **HistogramTransformation**.

The image is now ready to be used as a mask in its own right. Simply click to select the original image, go to **Mask** and click **Select Mask** (or press **CTRL M**). Select the duplicate image from the list and click **OK**. The mask will be in such a way that we are actually protecting the weak signal areas and this is not what we want, so go to **Mask** and click **Invert Mask** (or press **CTRL SHIFT I**) to invert it.

Now we are ready to attack the small scale noise in the background. Open the **ATrousWaveletTransform** tool. Like we did before, we will attack the small scale layers more and get progressively less aggressive as we go up in scale layer. Check to enable **Noise Reduction** for layers **1** to **4**. For layer **1**, set **Threshold** to **4.000** and **Iterations** to **3**. For layer **2**, set **Threshold** to **2.000** and **Iterations** to **2**. For layer **3**, set **Threshold** to **1.000** and **Iterations** to **2**. For layer **4**, set **Threshold** to **0.500** and **Iterations** to **1**. Once done, click the square **Apply** button.

The image has received an extreme improvement in noise level. We now close **ATrousWaveletTransform** and open **MultiscaleMedianTransform**. Check to enable **Noise Reduction** on layers **1** to **4**. Set **Threshold** to **4.000**, **2.000**, **1.000** and **0.500** respectively. Click the square **Apply** button once done.

Not a huge improvement in noise level but welcome nonetheless. We can now close **MultiscaleMedianTransform**. With the image well noise-reduced, we can remove the mask by going to **Mask** and clicking **Remove Mask**. The duplicate image we created can also be closed without saving. Save your reduced-noise Luminance image. I added *_NR* to the filename and moreover, save it in **64-bit IEEE 754 floating point** format.

Ensure you click **Reset** on **ScreenTransferFunction** and close the tool as we will now stretch the Luminance image. For this, open **HistogramTransformation** and here, click **Reset** and then click **Real-Time Preview** (so we can see what we are doing in real-time) and select the image from the list (so we can inspect its histogram as we stretch it).

Like we did before, all we need to do to stretch this image and enhance contrast is move the mid-tones slider toward the left. At the top you will notice the histogram widening and moving toward the right. As you drag the mid-tones slider to the left, make sure the histogram at the top does not get very far as data will be lost if it goes beyond the white point (the end on the right). The first stretch can be, and should be, very aggressive, as shown below.

Once you are happy with your first stretch, click the square **Apply** button and then click **Reset**. The background is now quite bright and so we need to clip the black point a bit. To do this, move the black-point slider to the right. In theory it should meet the very beginning of the histogram but because we are dealing with a DSLR image, and the background is therefore quite noisy, we cannot afford to do this (without making the background blotchy and unpleasant to look at). Therefore, clip the black point but do not go so far so as to reveal *blotchiness* in your background. See below for an example.

A second, much less aggressive, stretch would be good to bring out some further contrast. Just slide the mid-tones slider to the left a bit and at the same time, the black-point slider to the right a bit. Enough to achieve more contrast with a nice background (not blotchy or overly bright either).

Once happy with your second stretch, click the square **Apply** button and then **Reset**. Close **HistogramTransformation** and the **Real-Time Preview** window. Save your stretched image. I added *_HisT* to the filename.

After having amply stretched the Luminance image, it is time for some extra noise reduction because we move on to enhance fine detail. We will need to create a new mask to protect the strong signal areas and attack the weak signal areas. To do this, right-click on the image and click **Duplicate**. We will accentuate the weak and strong signal areas to make the mask more effective. To do this, open **HistogramTransformation**. Select the duplicate image from the list and click it to select it. Now click the buttons **Auto clip shadows** and **Auto clip highlights**.

Making sure the duplicate image is selected, click the square **Apply** button once done to prepare the mask. Close **HistogramTransformation** and apply this mask to the original image by going to **Mask** and clicking **Select Mask** (or press **CTRL M**). Select the duplicate image from the list and click **OK**. Again, the mask will need inverting to attack the weak signal areas. Go to **Mask** and click **Invert Mask** (or press **CTRL SHIFT I**).

ATrousWaveletTransform will again be used to produce what will now be a minor noise reduction. Open this tool, click **Reset** and then check to enable **Noise Reduction** only for layers **1** and **2**. Set **Threshold** to **3.000** and **Iterations** to **2** for layer **1**. Set **Threshold** to **0.500** and **Iterations** to **1** for layer **2**. Click the square **Apply** button once done.

We now close **ATrousWaveletTransform** and open **ACDNR**. Like before, we set **StdDev** to **2.0** and **Amount** to **1.0** (to apply stronger noise reduction and to not blend the denoised image with the original). Click the square **Apply** button once set.

With the background very nicely cleaned up, we close **ACDNR**. The mask can now be removed by going to **Mask** and clicking **Remove Mask**. The duplicate image we created as a mask can also be closed without saving. With some of the background blotchiness removed, we can actually darken the background a bit by moving the black point. For this, open **HistogramTransformation** and move the black-point slider to the right a bit (do not over-do it and check with **Real-Time Preview** to make sure background does not darken too much and become blotchy).

Once happy, click **Apply**, close **HistogramTransformation** and save your image. I added *_NR* to my filename.

What we now aim to do is increase the contrast in localised areas of the image, namely the strong signal areas. For this, we need to create a special kind of mask that will only include the stars and the brightest areas of the image. PixInsight has two tools we need to use to create this special mask - **StarMask** and **RangeSelection**. We begin with **StarMask**, so open this tool.

In this tool, we set **Scale** to **7** to include larger scale structures. We also set **Large-scale** to **0** so we are protecting small scale structures only and **Smoothness** to **8** (generally a good compromise for precision). One done, simply click **Apply** and the mask will be created.

We can now close the **StarMask** tool. The star mask image created can be minimised and moved to the side for later use as we actually need to combine this with another one to make our special mask.

We now open the **RangeSelection** tool.

Click the **Real-Time Preview** button and now slide the **Lower limit** slider slowly toward the right. You will notice less and less of the image staying white and more of it becoming black, i.e. we are excluding more and more and only keeping the brightest parts selected. You can toggle **Real-Time Preview** on and off to check precisely which parts of the image are being selected as you change **Lower limit**. This helps decide when you have chosen the important parts of the image - the areas that you wish to enhance (only strong signal areas should be selected!). Once you are happy with your selection, change **Fuzziness** to **0.15** and **Smoothness** to **5**. You may wish to make minor changes to your selection after this (perhaps lowering **Lower limit** to include more). Click the square **Apply** button once happy to create the image. Close the **RangeSelection** tool and **Real-Time Preview** window once done.

This mask image can also be minimised and set to the side. We will need to combine them and the best tool for that is **PixelMath**, so open this.

To combine both masks into one, type the following in **RGB/K**:
range_mask+star_mask

This will literally add both images together. Expand the **Destination** tab and check to enable **Create new image**. Ensure **Rescale result** is *not* enabled as we do *not* want to rescale. Click the square **Apply** button once done.

We can now close the **PixelMath** tool. This is our special mask, but to make the masking smoother, we will blur the image heavily. With this new mask image selected, open **ATrousWaveletTransform**. Click **Reset** and then for layers **1** to **4**, uncheck to disable **Detail Layer** on each. Click the square **Apply** button *three times* once done (to apply this blurring three times over).

Close **ATrousWaveletTransform** and now apply this blurred image as a mask to our original image. To do this, go to **Mask** and click **Select Mask** (or press **CTRL M**). Select the image from the list (do not select *star_mask* or *range_mask* as these are not the image we just created!) and click **OK**.

You will immediately notice this mask is unlike the others we used earlier. This mask is more *specific*, selecting out the strong signal areas. What we will first do is enhance contrast on very large scale structures. This is commonly very diffuse nebulosity found all around the image. Since we have heavily reduced noise in the background, it is fairly safe to do (but be careful as DSLR images do tend to have a lot of background noise!). Let us first invert the mask to protect the strong signal areas (as we will try to enhance contrast on low signal areas). To do this, go to **Mask** and click **Invert Mask** (or press **CTRL SHIFT I**). Once done, open the **LocalHistogramEqualization** tool.

Kernel Radius here represents the pixel scale we wish to enhance. Since we are talking about very large scale nebulosity, we enter a value like **200**. **Contrast Limit** is an important parameter as this sets by how much to enhance contrast. A setting of **1.0** does nothing and anything above that enhances contrast accordingly, but be careful as the higher this is, the more noise it can introduce and bring out. Generally a setting of **1.5** to **2.5** is used. In **Amount** we can set a blend between the contrast-enhanced result and the original image. **1.000** means we keep the contrast-enhanced image entirely and anything below is a blend thereafter. For **Histogram Resolution**, **8-bit (256)** works well but anything higher will take longer to process but in theory be more accurately calculated. For my image, I set **Kernel Radius** to **200**, **Contrast Limit** to **1.2** (to avoid too much noise), **Amount** to **1.000** and left **Histogram Resolution** at **8-bit (256)** (as at extremely large scale, higher accuracy is not important).

To see the results, it helps to hide the mask. To do this, just go to **Mask** and click **Show Mask** (or press **CTRL K**). The mask is still present and enabled, but it is hidden. This can be toggled on and off. If you are unhappy with the result, just press **CTRL Z** to undo and try different values.

Once happy, invert the mask (go to **Mask** and click **Invert Mask** or press **CTRL SHIFT I**) and let us now enhance contrast in the strong signal areas. This will again be done with the same tool, **LocalHistogramEqualization**. This time we set a much lower **Kernel Radius**. **32** to **64** is generally good for smaller scale detail. Avoid going too low as it gets into the scale domain of noise. As these areas have significantly stronger signal, you can afford to use a higher value of **Contrast Limit** and it may also be a good idea to set a higher value of **Histogram Resolution**, such as **10-bit (1024)** or **12-bit (4096)**. For my image, I set **Kernel Radius** to **48** (as it is a very wide-field image and fine detail is very small), **Contrast Limit** to **1.5** and **Histogram Resolution** to **10-bit (1024)**. **Amount** was again left at **1.000**.

This has provided tons of enhancement to contrast in areas of nice fine detail. Since we stretched our image extensively earlier, we would be able to recover some of the detail with a little bit of HDR treatment. This is optional really, and it will work better in some images than others. The **Great Orion Nebula** is a good target for this treatment as it saturates quickly. The tool for the job is **HDRMultiscaleTransform**.

In this tool, we first need to set the limit of pixel layers to attack. The lower we set this, the stronger the effect is as we only end up doing it to smaller scale layers. A setting of **5** is a good value for **Number of layers**. One iteration suffices so we can leave this alone. We do want to apply this to lightness however, so we check to enable **To lightness**. Go ahead and click **Apply** and see what happens to the strong signal areas.

The mask has been hidden above to show the detail clearly. Much of the fine structures within the **Great Orion Nebula** have been revealed with this tool as it has brought out the HDR in it. One of the reasons this tool was so successful is that we have been using *64-bit* format for the image since before we stretched it (maintaining the large range of values for contrast). Nevertheless, it is important we look at the bright stars after this tool has done its job, to make sure the star centres are not darkened or worse, black. The **Deringing** feature can be enabled to attempt to prevent this. Otherwise, altering the **Number of layers** can help (higher number). Once we are happy, we can close **HDRMultiscaleTransform**.

One final enhancement we can apply, also optional, is a tiny bit of sharpening to fine details within strong signal areas. For this we use the **ATrousWaveletTransform** tool. Click **Reset** on it once opened. The way we sharpen is simply to apply a little bit of **Bias** to layers. We avoid layer **1** entirely as this layer can be dominated by small scale noise. Please note this can be very quickly over-done. My recommendation is to apply a little bit of **Bias** to layers **2** and **3** (maybe **4** but just as a test). Start small, at say **0.050** for layers **2** and **3**.

There is a noticeable improvement to fine detail sharpness in the image. Do make sure you check every part of the image and see whether or not this has introduced too much noise, looks unrealistic or simply over-does it (stars may appear too accentuated). **0.050** seems to be a good value in my experience, but playing around is key. Once happy with your image, close **ATrousWaveletTransform**. Remove the mask by going to **Mask** and clicking **Remove Mask**. You should also close the *star_mask*, *range_mask* and special mask images (without saving any). Make sure you save your image. I added *_Enhanced* to the filename.

STEP 7. Combining our Luminance and RGB colour images

After having extensively post-processed the RGB colour and Luminance images separately, we are ready to combine them together. To do this, first close every image in PixInsight and then open only the finalised RGB colour and finalised Luminance images.

The tool for the job is **LRGBCombination**, so open this tool.

We will essentially get our Luminance image and apply it to our RGB colour image. To do this, click to select the RGB colour image and uncheck to disable **R**, **G** and **B** from **LRGBCombination**. From the list for **L**, select the Luminance image. Expand **Channel Weights** and make sure everything is set to **1.00000**. Check to enable **Chrominance Noise Reduction** (and leave at default settings as these work well - this bit is to reduce noise produced by the LRGB combination process). In **Transfer Functions**, we should pay attention to **Saturation**. The default value is **0.500** and this means that no boost to colour saturation is provided by the LRGB combination process. If this value is decreased, to say **0.300**, colour saturation is boosted. If the value is increased, to say **0.700**, colour saturation is lowered. I tend to set a value of around **0.400** and check the result. If it looks good, I undo, lower to **0.350** and try again. I keep doing this until colour saturation looks pleasing and not over-done. To perform the LRGB combination process, click the square **Apply** button but do make sure the RGB colour image is selected first.

The genius behind the whole post-processing ordeal of separating Luminance and RGB to later re-combine comes into mind as soon as you see the result. By splitting up Luminance to post-process meticulously and bring out masses of detail, after having blurred and denoised the RGB image immensely, you get an extremely pleasing result that is both rich in colour and rich in detail. Your resulting image will of course depend on all sorts of factors, including whether or not you carried out the steps above mentioned as optional. At this stage, once happy, close **LRGBCombination** and your Luminance image. Save your new colour image (after LRGB combination) as a new file. I saved mine named as *LRGB*.

STEP 8. Final enhancements to contrast and colour saturation

Before we end this tutorial with the final result, we can make some global improvements to the image to bring out the contrast and colour saturation further. This is done with the **CurvesTransformation** tool.

Click the **Real-Time Preview** button to see what we are changing in real-time. With the **RGB/K** mode selected, we increase contrast by making an S-curve. To do this, click to add a point on the first quarter that is lower than the line. Then, click to add a point on the third quarter that is higher than the line. This will effectively make an S-curve in **RGB/K** and enhance contrast on brighter areas (to bring them out). For DSLR images, it helps to exaggerate the S-curve a little (especially to drown out some of the background blotchiness from noise).

There is a strong enhancement already, with the background more tamed and the nebulosity less washed out. Since this has the effect of darkening the image, we click the **L** to go into lightness mode. Increase lightness only slightly. To do this, add a point in the middle and drag it up slightly toward the top-left (this curves the line).

Now to give colour saturation a bit of a boost, in a very similar way. To do this, click the **S** to go into saturation mode. Increase colour saturation by doing what you did with lightness - add a point in the middle and drag it up toward the top-left to curve the line. For this, do it a bit more aggressively than for lightness.

To see the difference achieved, it helps you click the **Real-Time Preview** button on and off as this will quickly toggle the changes on and off. Once happy, click the square **Apply** button to apply the changes the image and then click **Reset** to see the image as it is at this time. Feel free to play with this tool a little more (a second iteration of the above, perhaps), but do not be as aggressive the second time.

A less aggressive second iteration proves to provide a nicer overall final result. Once completely done, save changes made to your image and publish your wonderful result! Do remember that your final image is still in *64-bit* format. To publish your result, choose to save in **16-bit unsigned integer** for **TIFF** to maintain excellent colour. **JPEG** is decent for posting on forums and such, but try to keep to **100** for quality.