Comparison of processing JPEGs and FITS/RAW files

This is an experiment to see exactly what the difference between processing images that have been compressed using lossy compression routines (JPEG) and images that are in a lossless format (e.g. FITS, .tiff)

What are JPEG files?

JPG files, also known as JPEG files, are a common file format for digital photos and other digital graphics. When JPG files are saved, they use "lossy" compression, meaning image quality is lost as file size decreases. JPEG stands for Joint Photographic Experts Group, the committee that created the file type. JPG files have the file extension .jpg or .jpeg. They are the most common file type for images taken with digital cameras, and widely used for photos and other graphics used on websites.

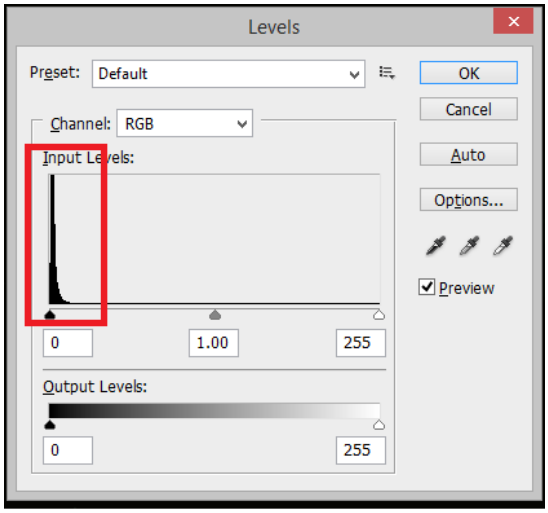
What is lossy compression?

In information technology, "lossy" compression is a data encoding method that compresses data by discarding (losing) some of it. The procedure aims to minimize the amount of data that needs to be held, handled, and/or transmitted by a computer. Typically, a substantial amount of data can be discarded before the result is sufficiently degraded to be noticed by the user.

What is astro-photography?

The aim of deep sky photography is to show faint objects. These objects are faint, typically much fainter than the naked eye can see. In the past astro-photographers used very long exposures to expose film emulsions. With the rise of digital photography we now can use digital sensors to record files that can then be manipulated using computers. Ultimately the aim is to enhance the faint detail that lies in the photograph.

When we take an image with a camera the image data is said to be “linear”. That means that the data within the image has not been “stretched”. If you look at a histogram (a graphical representation of the data contained within the image) you will see that the histogram has a peak on the left of the graph:



This means that all of the data is crammed into the darkest parts of the image. In a histogram the darkest pixels are on the left (near the “black-point”) and the brightest are on the right (the “white-point”). In the example that I used below, you can see that the image is very dark, with only a few bright stars and the core of M42 visible.

The process of processing astro-photographs aims to “stretch” this data so it become wider and wider. This brightens up the data contained within the dark areas, making them visible.

So why is it important to use RAW/FIT files for astrophotography?

When an image is compressed using lossy compression routines, the compression algorithm takes areas of uniform colour and replaces them with a single piece of information. The algorithm doesn't care that the very data that we are trying to enhance may lie within those areas of seemingly uniform colour. In effect it disposes of the very data that we are trying to enhance!

What is an uncompressed file type?

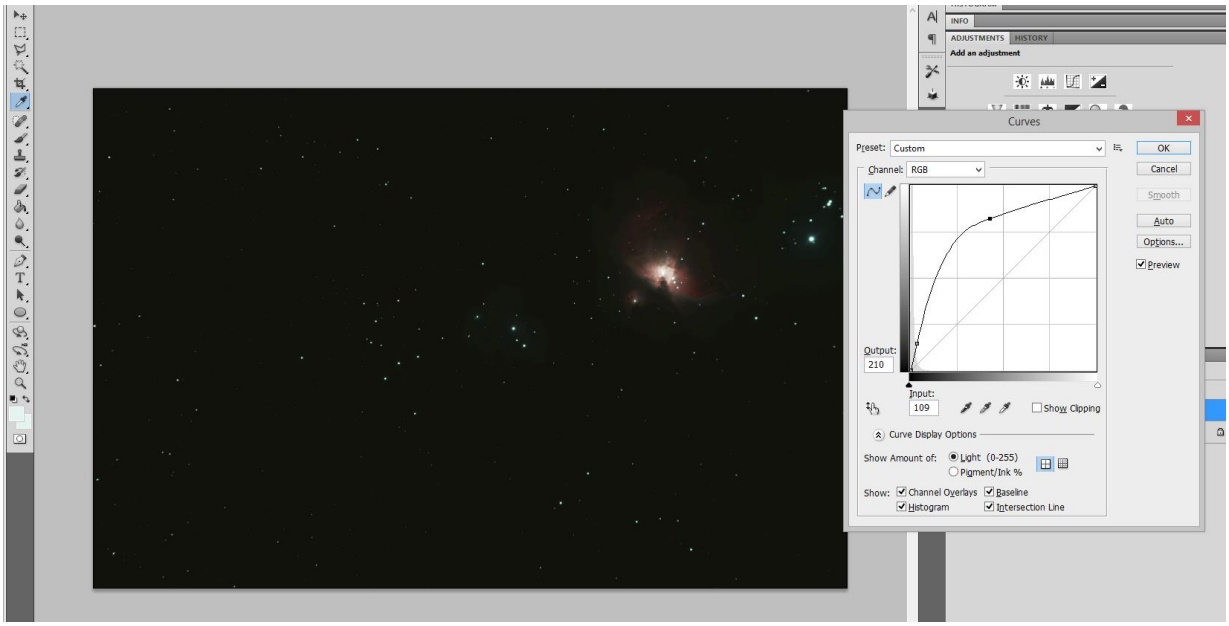
There are many different types of image files that have not been compressed. The ones that you will become familiar with in astro-photography are RAW (the unprocessed output from a DSLR), TIFF (*Tagged Image File Format*- an image file that can be opened by most computers) and FITS (*Flexible Image Transport System*- an image format that is output by astro CCD cameras). These files do not dispose of any data, and as such are much larger than JPEG files.

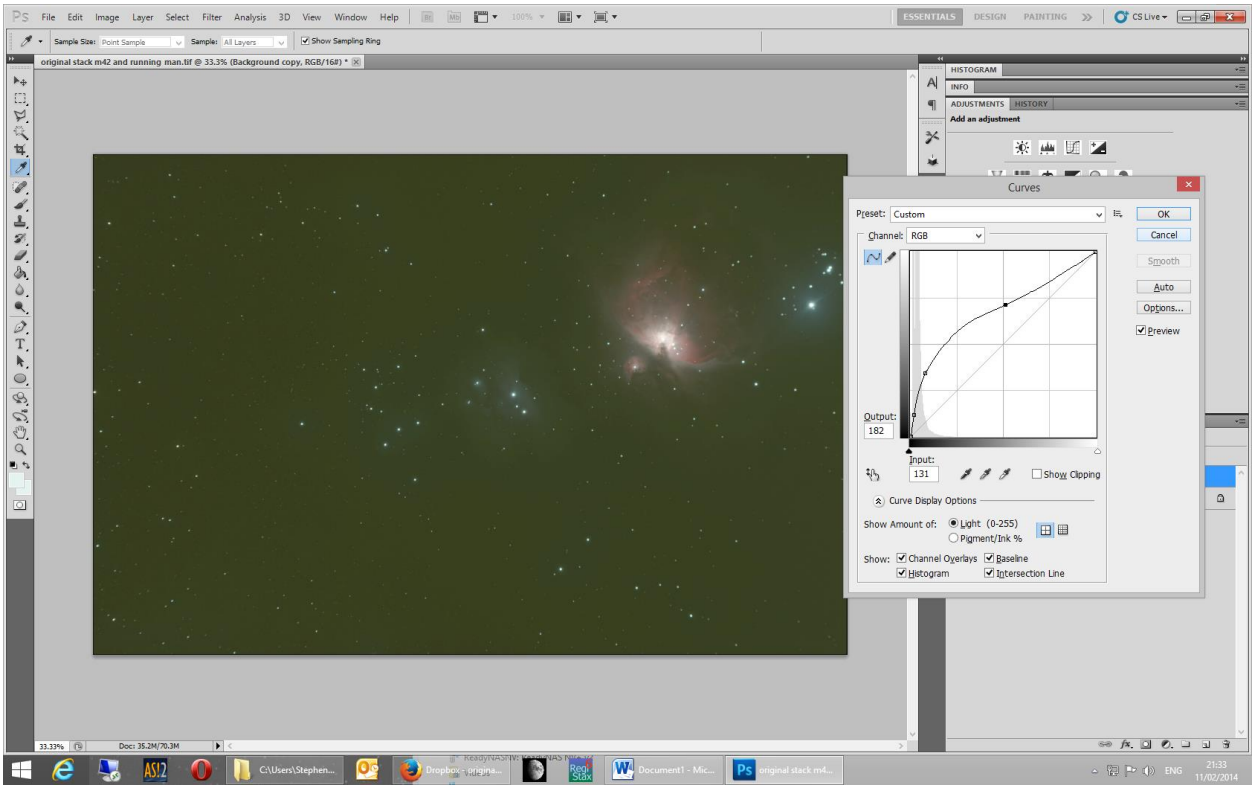
The processing workflow using a 16-bit .tiff file

This is a stack of RAW files that have been stacked in Deep Sky Stacker.

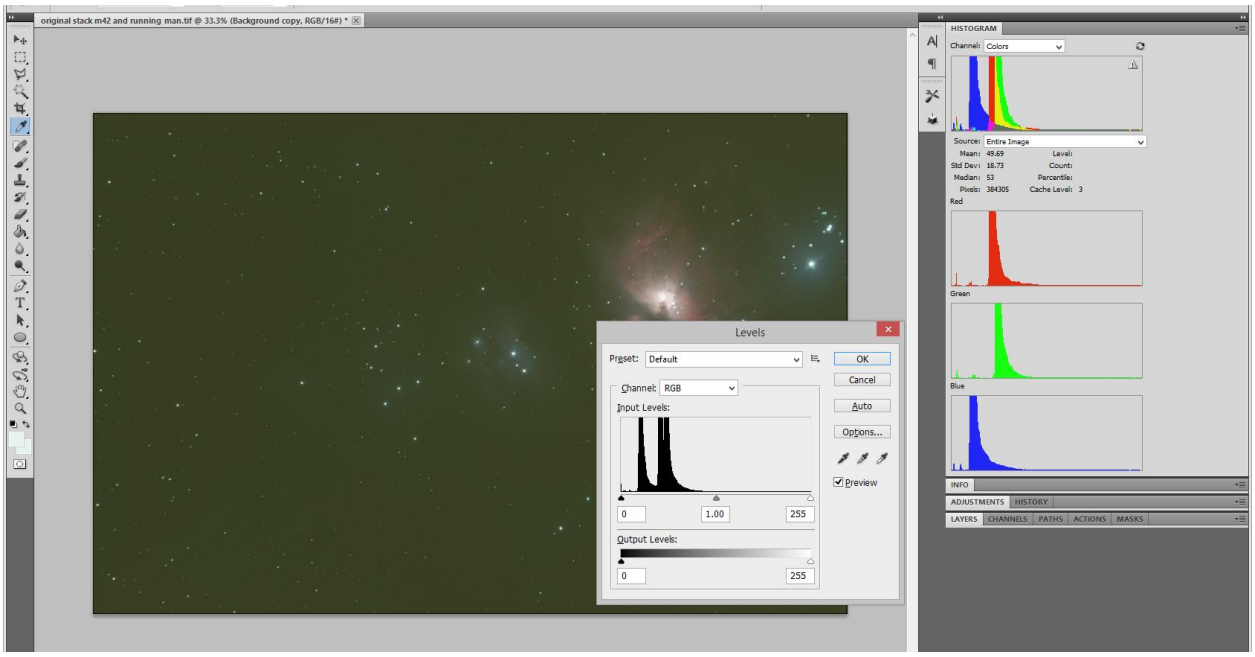
No processing took place in DSS. The stack was saved as a 16 bit .tiff file.

A number of initial histogram stretches were carried out:

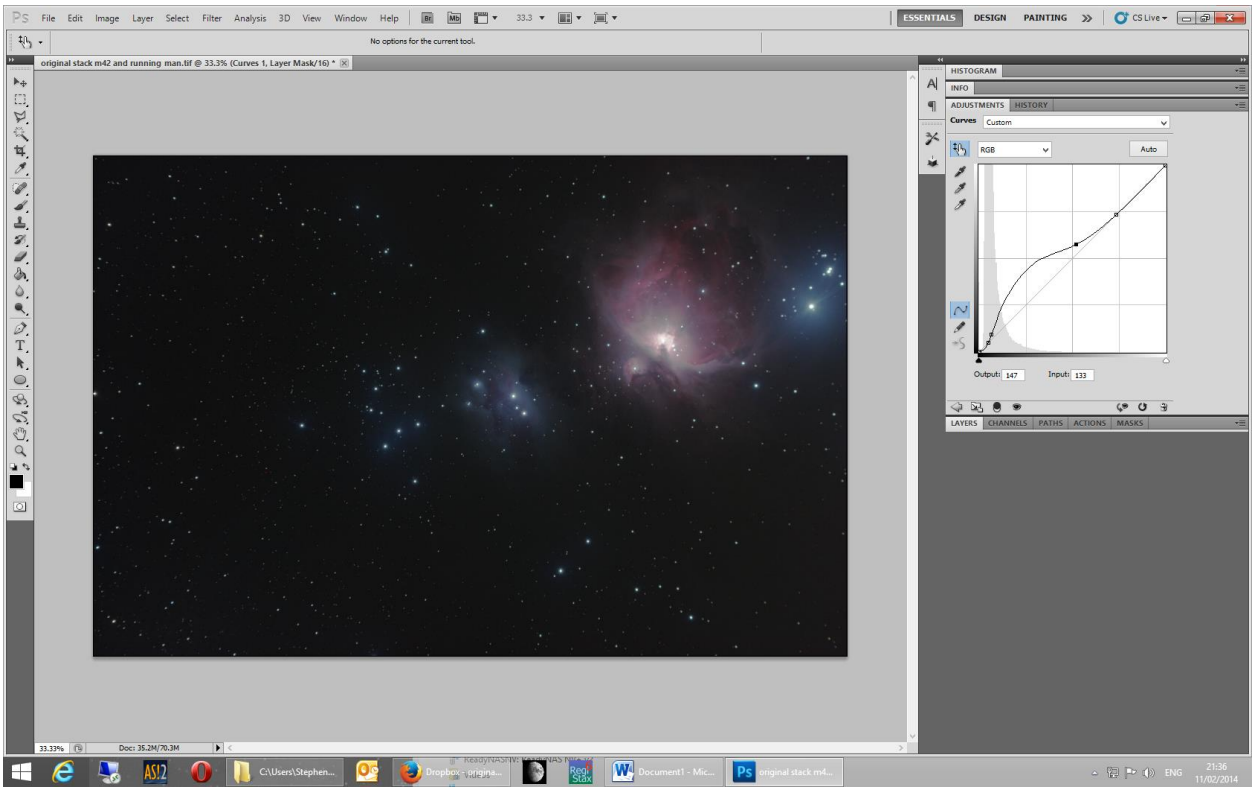

Curves
Preset: Custom
Channel: RGB
OK
Cancel
Smooth
Auto
Options...
Preview
Output: 210
Input: 109
Show Clipping
Curve Display Options
Show Amount of: Light (0-255)
Pigment/Ink %
Show: Channel Overlays Baseline
Histogram Intersection Line


Curves
Preset: Custom
Channel: RGB
OK
Cancel
Smooth
Auto
Options...
Preview
Output: 182
Input: 131
Show Clipping
Curve Display Options
Show Amount of: Light (0-255)
Pigment/Ink %
Show: Channel Overlays Baseline
Histogram Intersection Line

The individual channels were then aligned using Levels:

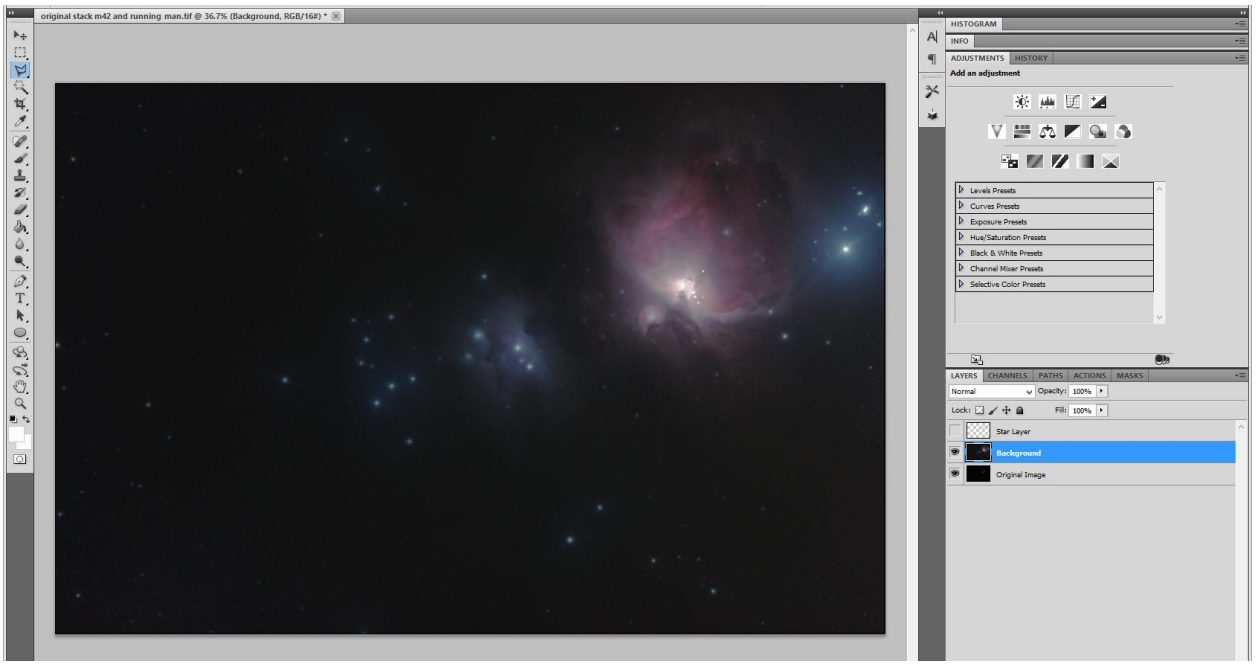


Another iteration of histogram stretching



At this point I created a star mask to protect the stars from bloating. This allow the background to be stretched to bring out the detail without blowing the star shape.

This leaves me with a star layer and a background layer. Most of the processing now happens in the background layer



From here I carried out more histogram stretches. I then sharpened the gas clouds using a High pass filter, which was layered in. The stars were then layered back in.

The final image ended up looking like this:

All of the now-visible nebulosity was contained within the original dark image.

Using JPEG

The same original image was converted into a JPEG file in Photoshop. The largest, most detailed format was used to ensure that the minimum of data was disposed. The original file was 35Mb in size. The JPEG file was 149Kb in size. In other words the original image was 241 times larger than the JPEG. This shows just how much data was disposed by the compression routine.

Before processing though, there is very little difference to the eye in to two images.

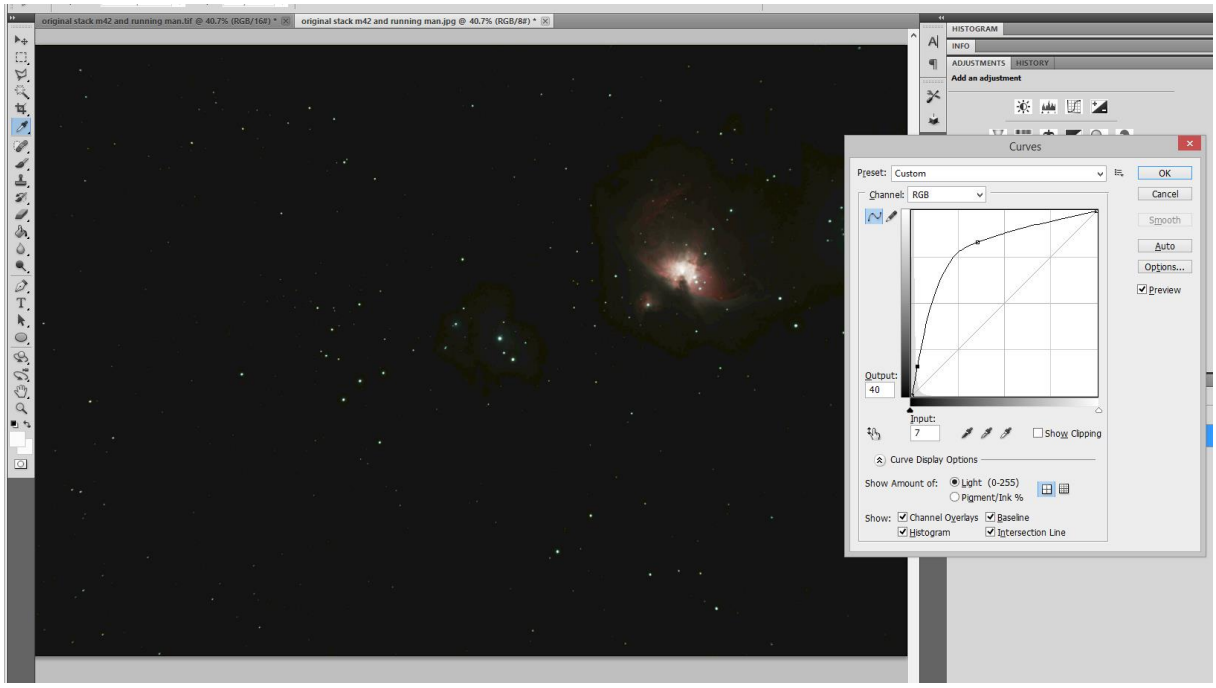
Here's a blown-up version of the original TIFF file:

And here's the same region blown-up in the JPEG file:

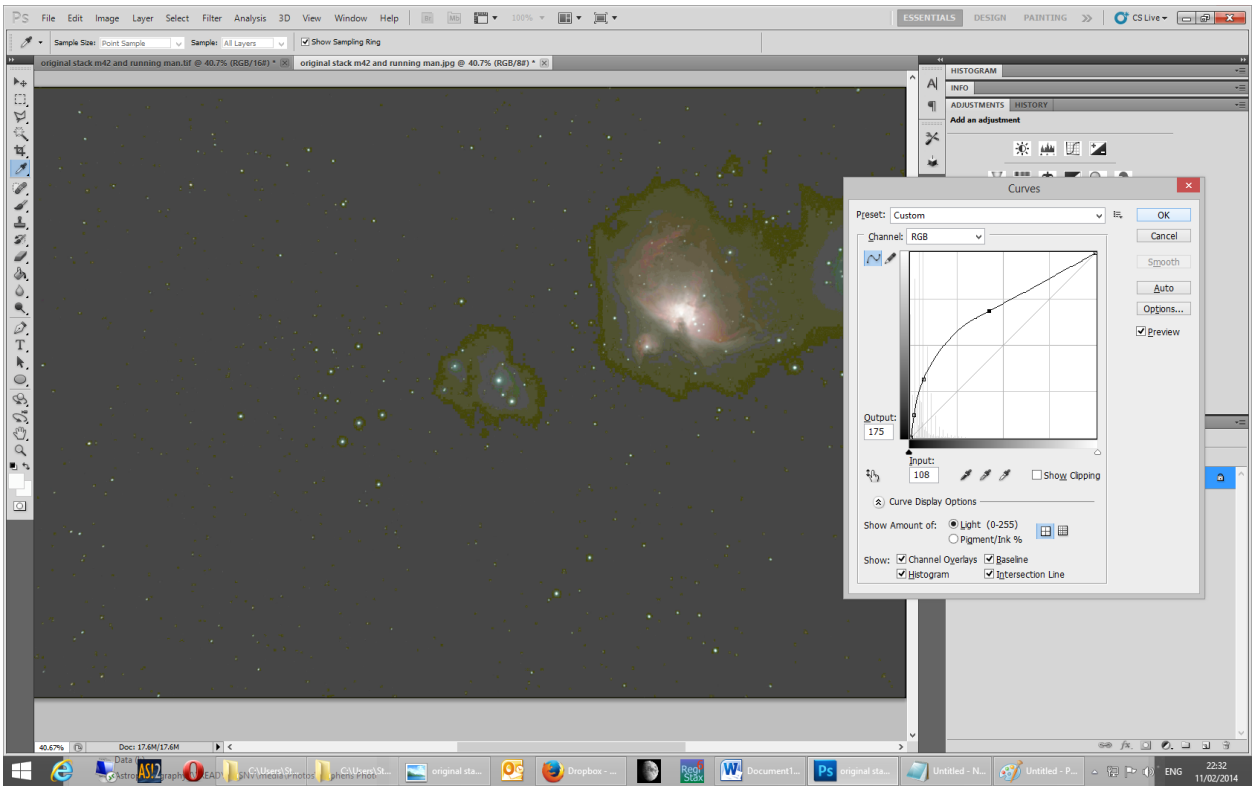
Both images look identical to the eye, which just goes to show how good the compression routine is at getting rid of the data that won't be noticed.

However, let's try to stretch the JPEG image:

Notice how the first stretch using the Curve tool is almost identical to the first stretch that was applied to the .tiff file? The image looks OK (ish) but if you look closely you can start to see patchy areas of colour appearing around the nebulae

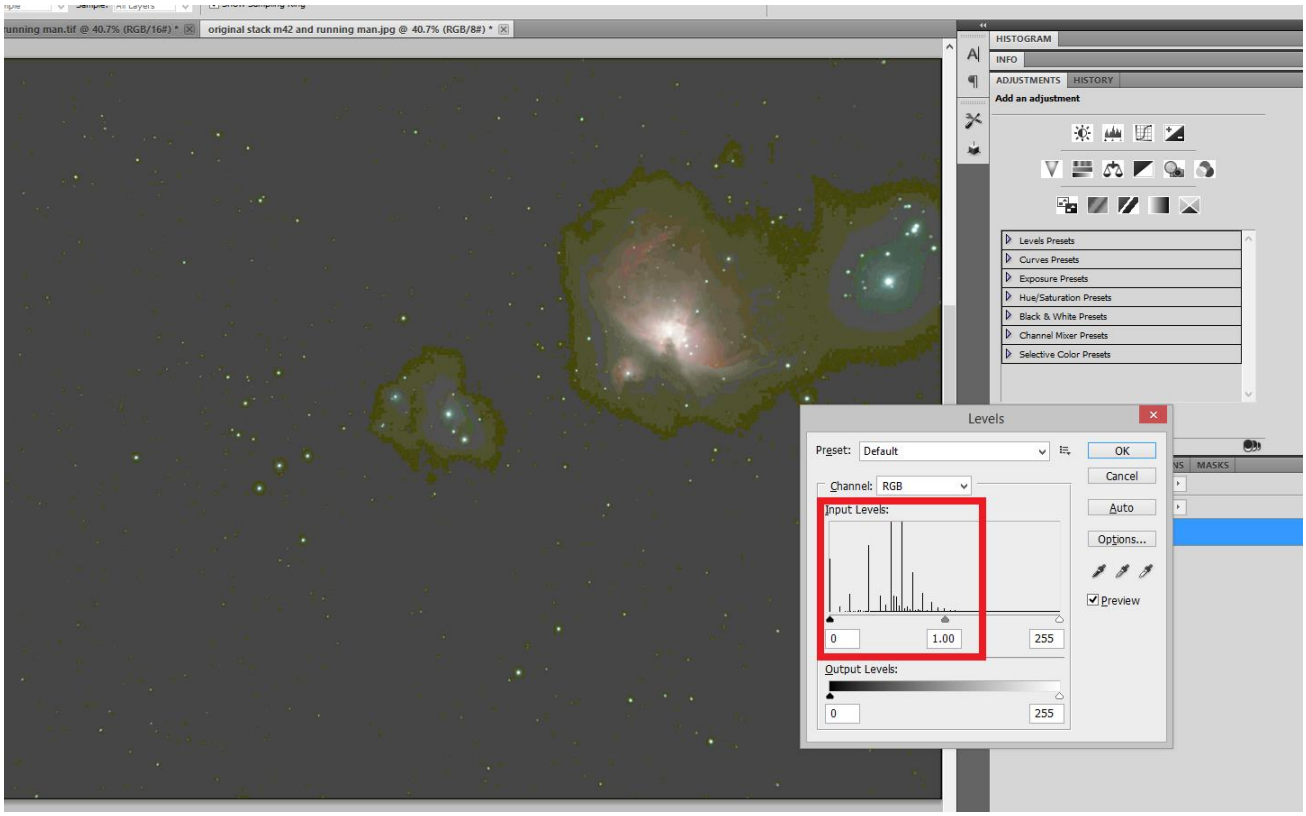


Let's apply a second stretch (again, identical to the second stretch in the original file):



Now things have taken a massive turn for the worst, with huge patches of blocky colour appearing.

If we look at the histogram we can see that it is now composed of jagged peaks. This is because there isn't enough data to fill the histogram and the image is starting to break down.



At this point there really is point in proceeding any further. The image simply does not have enough data contained within it to try to process it. All of the detail that we are looking for has been disposed by the JPEG compression routine.

Experiment 1 Summary

JPEG files are great for displaying the finished article as the image is much smaller than the original. However the process of compressing a file to JPEG throws away the very data that we are trying to enhance.

You have worked hard to capture the ancient photons from distant astronomical objects. Don't throw that had-won data in the bin before you have processed it. Do not convert your data to JPEG until you have finished processing it!

Notes and Assumptions

This experiment applied the JPEG compression after the original images had been stacked in Deep Sky Stacker. The original files that were stacked were in FITS format (as output from a QHY8L CCD camera). This means that the best possible quality came from the stacking process. I may try to re-run this experiment but use DSLR images. These would have to be converted from RAW to JPEG and then stacked. I would expect the results to be worse, as DSS would be stacking compressed files.

Part 2:- Comparison of stacking JPEG vs RAW

In this section I will compare the effects on image quality when stacking JPEGs compared to stacking RAW images.

For this session, I used data that I collected with a Canon 50D. The images were 80 second exposures, taken on an unguided wide-field imaging rig (Kenko Skymemo mount, Canon 50D, 70/200f4L lens). A total of 36 sub-exposures were used, totalling 48 minutes exposure time. The images were captured in RAW format (Canon CR2 files). To produce JPEGs I batch-converted the RAW files to JPEG using IrfanView's batch-conversion process, with the JPEG output set at the highest level of quality. The CR2 files were 20mb each and the resulting JPEG files were 4.6Mb each.

This is a sample of the sub-exposures

The JPEGs and CR2 files were stacked in Deep Sky Stacker, using a Median Kappa-Sigma stacking routine. No calibration files were used.

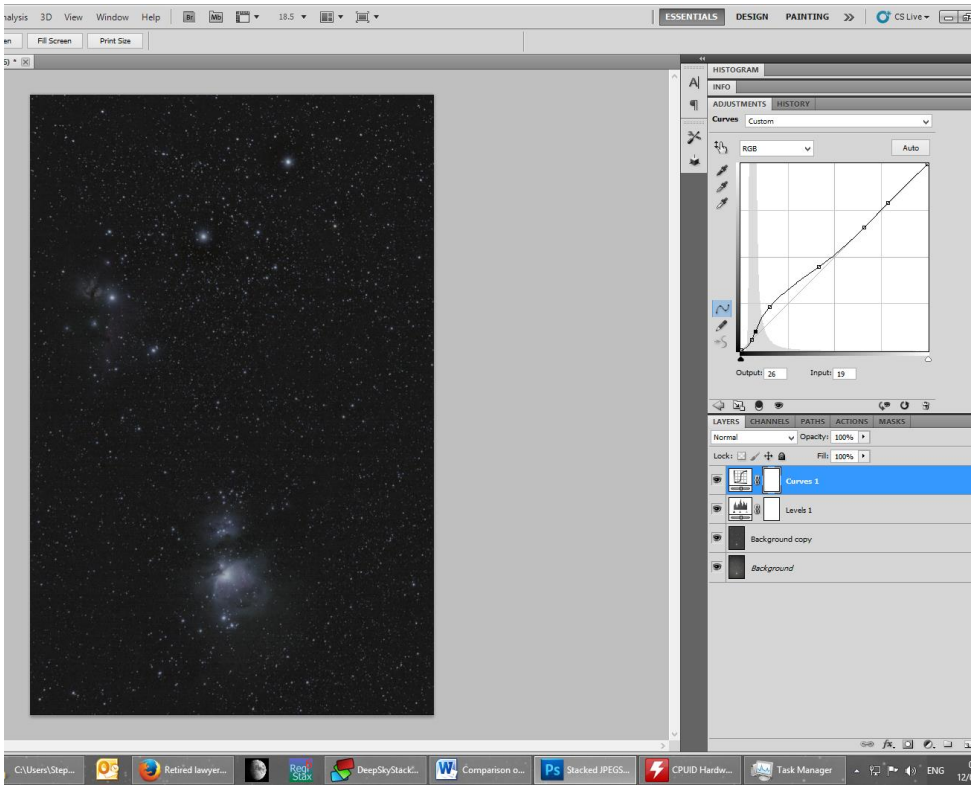
The first thing that I noticed was that DSS was having trouble identifying stars. I checked in the Advanced section to see how many stars DSS was using to register the files (you should aim for around 100 stars). At 41%, DSS was reporting 2500 stars. At 44% it was reporting 0 stars detected. The best compromise that I could find was 43% which resulted in 354 stars being detected.

Normally, when stacking RAW files, DSS has generally no problem detecting stars as long as the star shape is good. If the stars are oblong from poor tracking, or if the stars are badly trailed then DSS's quality routines will reject the stars. What can happen then is that DSS will try to lock onto what it thinks are stars. Very often it will end up locking onto noise within the image. When this happens it will report seeing thousands of stars. DSS will fail to stack as it cannot register images using noise, as the noise is random in nature.

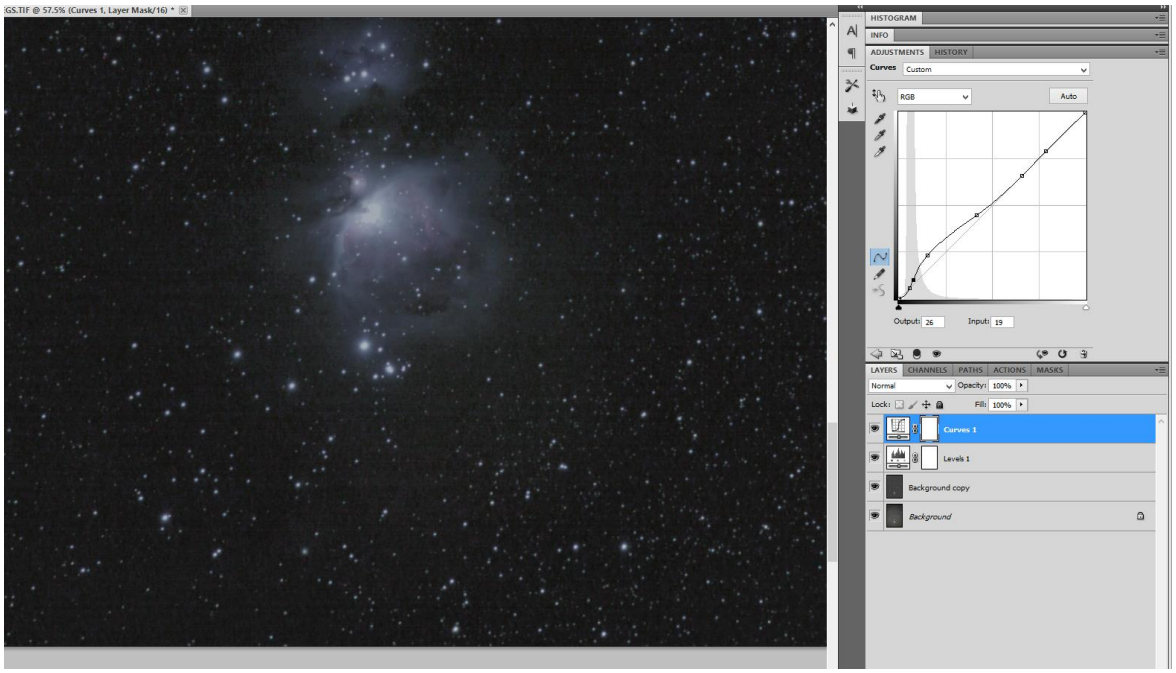
As to why DSS seems to have greater difficulty identifying stars in these JPEGs? My theory is that by compressing the image into JPEG you have disposed of data in the dark region. This may make it harder for DSS to identify stars as their brightness does not tail off smoothly, which may make it harder to identify the star centroid. I confess to not knowing enough about the inner workings of DSS to be able to fully qualify this hypothesis.

JPEG Processing

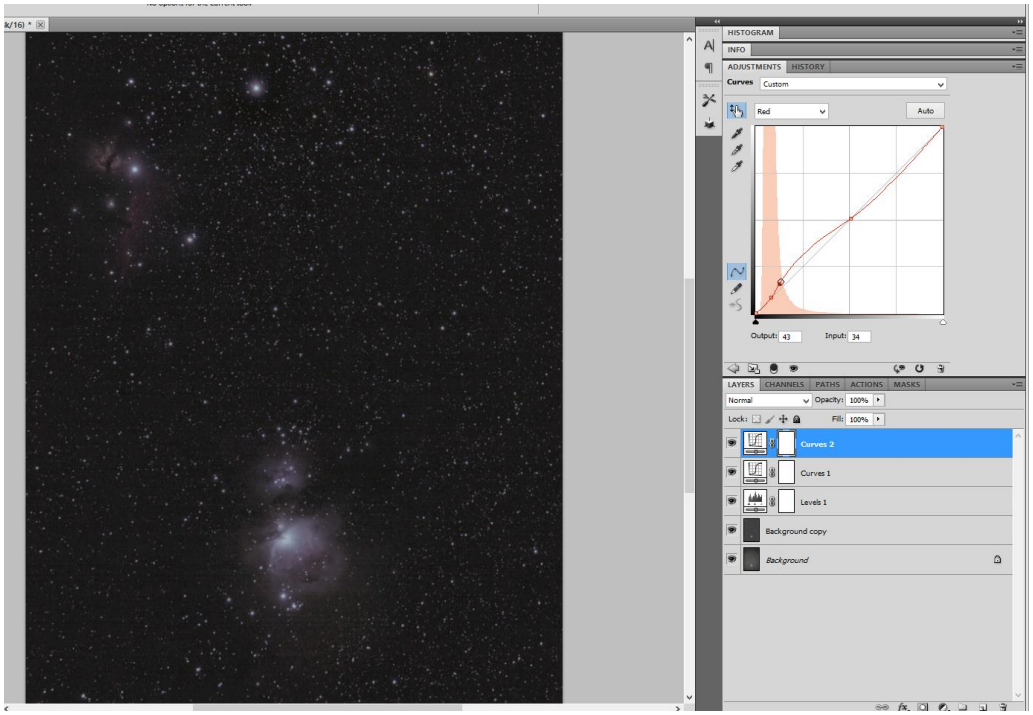
First Stretch



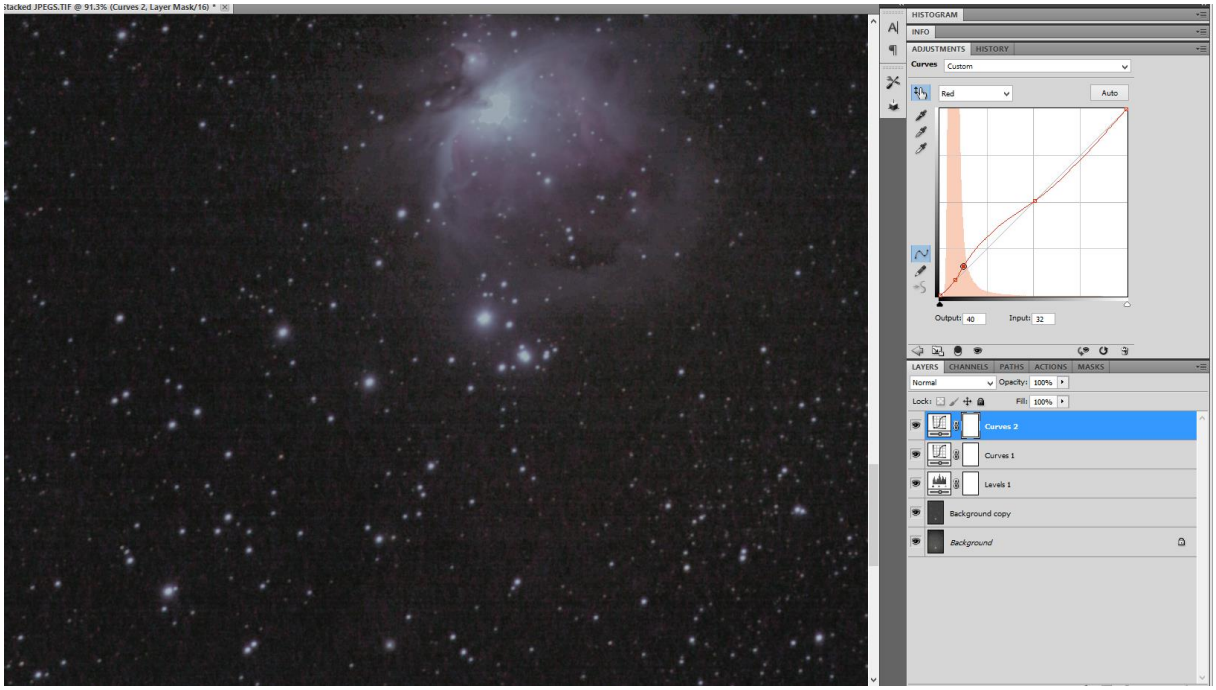
Noise after first stretch



Red data stretch



2nd Stretch Noise



Before and After Processing

RAW File Processing:

Before and After Processing:

I have not detailed the processing steps that I took in handling the 16-bit uncompressed data. Broadly speaking I used Levels, Gradient XTerminator, Curves (as adjustment layers) and then some noise-reduction.

Experiment 2 Summary

The difference was not as dramatic in the second experiment, probably because of the nature of the image (widefield with not much fine nebulosity). Therefore the stretching of the data was much more limited, which plays into the favour of JPEGs.

What was apparent that it was much easier to control the noise in the image that was uncompressed. The background noise was very difficult to control in the JPEG image and I had to do a number of noise-reduction routines, including reducing noise in each individual colour channel. My conclusion is that the post-processing of the JPEG image was much harder to get a result that doesn't look as good as an image created with un-compressed data.

I also note that no calibration files (darks, flats, bias) were used, as I specifically wanted to see the difference in the data. Using calibration files would have helped enormously with controlling the noise in the background. I also had to use a Photoshop plugin, Gradient XTerminator, to remove the very evident vignetting. Flats would have calibrated the image meaning that I probably could have left this step out. The less processing that you have to do on an image means that noise has a smaller chance of becoming an issue. Ultimately, more integration time and calibrating the image would have made the processing easier. The un-compressed data would still return a better result as the data can be pushed and stretched further.

In terms of speed of stacking and processing demands, stacking the JPEG images was a little faster. It took 8 minutes 15 seconds to stack the JPEGs and 9 minutes 40 seconds to stack the CR2 RAW images, which is about a 17% difference in time. So stacking JPEGs isn't that much easier on the PC requirements (I stacked on a 6 core AMD machine with 8Gb of RAM). The stacking used about 700Mb of RAM in both cases.

The quality of the data that you start off with has an incremental effect on each step of the journey to the finished image. If you start off with lower quality data, then at each step you have to work harder and harder to try and extract the detail. The process of astro-photography is not easy. Acquiring the data can sometimes seem like form of torture (guiding, mounts, software, CCD vs DSLR and so-on). Post-processing can seem like a black art, especially to a complete beginner.

However, starting the process off with poorer quality data means that you are making every subsequent step harder and harder.