Effects of noise in BIAS, darks, and flats and how to deal with them

Noise is an intrinsic random property of any taken image. It can be caused by a variety of factors, such as heat, long exposure times or high ISO settings. However, due to its random nature it can be averaged out if several images are taken. In this short report I explore the different methods used to reduce noise in astronomical images, particularly in BIAS, darks and flats, and how the different methods affect performance and results.

Noise in an image – how does it look like?

If we take any image from our telescope and visualize it, we can see that there is some randomness to it – that's what we call the noise. Take Figure 1, for example, which is a bias frame from the night of the RV UMa observations. This noise can be reasonably assumed to be a Gaussian distribution around the actual value of the bias, and the objective is to reduce the spread as much as possible to end up with a master file that is close to the real value of the bias in our camera. The same logic applies to dark images, and while flats technically follow a Poisson distribution, it becomes indistinguishable from a Gaussian at large enough values. Our images have $1.6 \cdot 10^7$ pixels, which is large enough for this limit to apply.

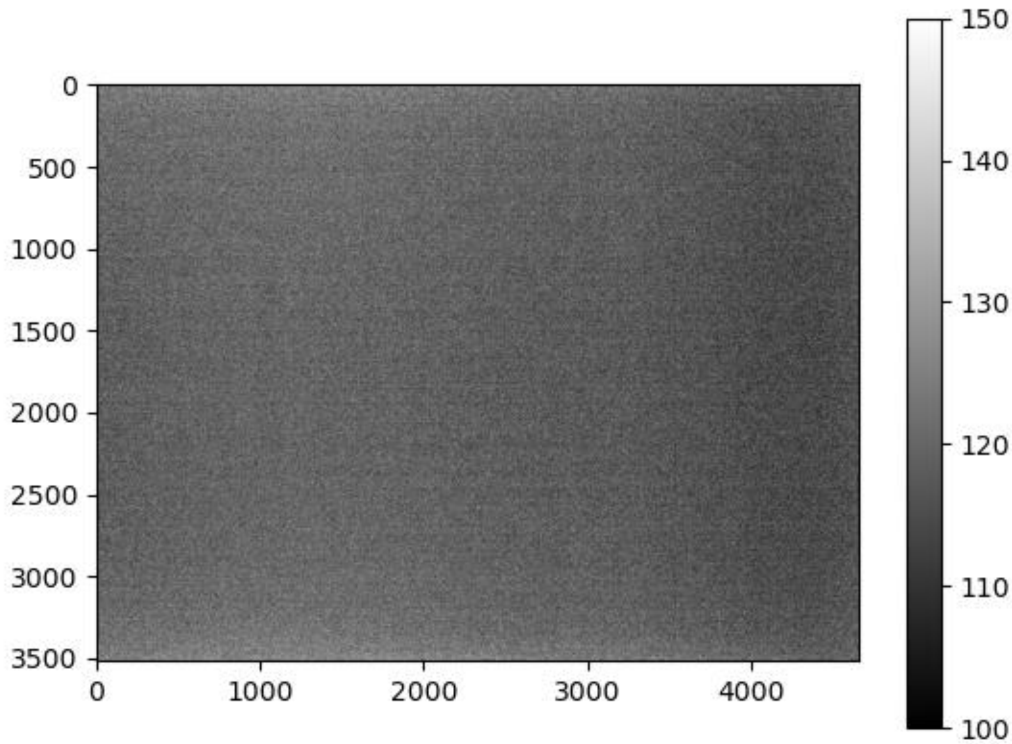

Figure 1: Dark image from the night observations of RV UMa.

If we plot all the other bias, we will see that the noise changes in each picture. We are expecting the BIAS in a certain pixel to be the average of all these values, which will remove the noise and leave us with the constant offset we are looking for. A master BIAS is just that, an average of all the BIAS files that helps us get rid of the noise so we can apply it to the science image.

To do so, we can just take the value of each pixel in each image and average them. Figure 2 shows this process done with a pixel in the BIAS files. It also shows the standard deviation and the standard error of the mean in the picture. We can see that both are small, and that the values of the pixels are all similar. In this case, just using the average works well, with the median and the average both having similar values.

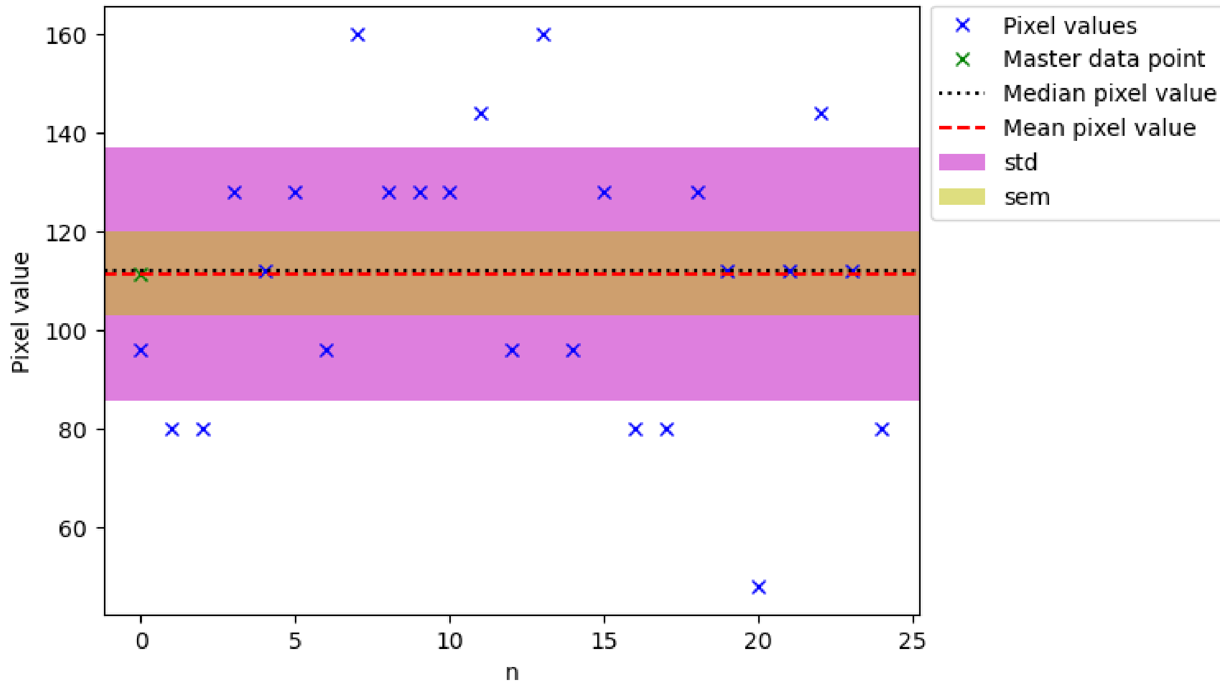

Figure 2: Pixel count for a random pixel in each of the darks for RV Uma. In this case the datapoint used in the master (green cross) is similar to the mean, but this is not always the case.

Outliers and their effect on the master

But what happens when there is an extreme outlier, which could be caused by a cosmic ray hitting the detector during the measurement. This is shown in Figure 3 taken from the Astropy data reduction guide, which shows a heavy outlier and how it affects the average and the median.

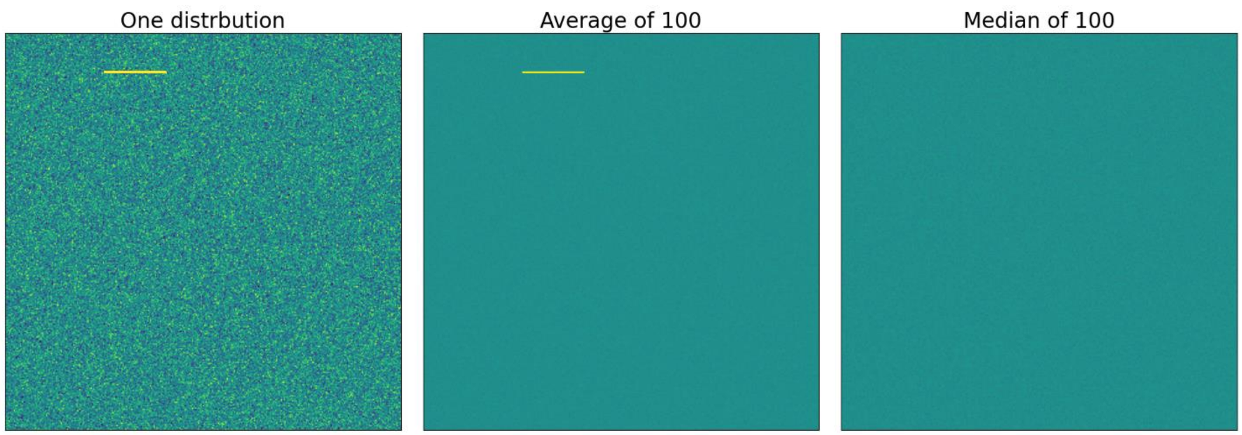

Figure 3: Effect of outliers on average and median. A single frame with a heavy outlier can significantly impact the average, but the median remains largely unaffected.

As can be seen, the average still has the outliers present, but the median does not. However, as can be seen in Figure 4 combining by average gives a narrower distribution of values around the median. It also takes about 10 times less to compute the average (2.59 seconds for the median, 266 miliseconds for the mean). Ideally, a combination of both methods should be used, the median to prevent outliers and the average to reduce the spread of the noise.

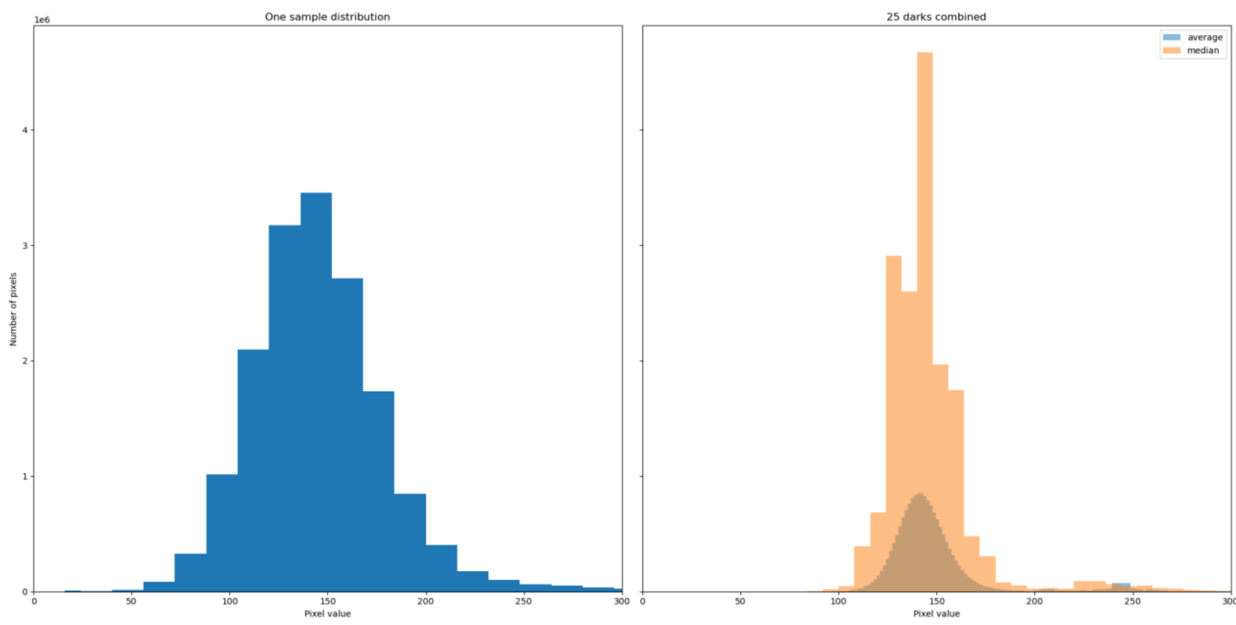

Figure 4: Distribution of values after taking the average and median of the bias files. Both methods significantly reduce the spread of values, but the average has the narrower spread of the two.

σ – clipping

The method used in Astropy, and the one we are using for this project, is called σ-clipping. It is defined as the following iteration:

1. Calculate the standard deviation (σ) and median (m) of a distribution.
2. Remove all points that are outside of the range $m \pm \sigma$
3. Rerun the iteration until the exit condition is met.

The exit condition we are using in this project is a tolerance: we are looking for the following condition to be met:

$$\frac{\sigma_{old} - \sigma_{new}}{\sigma_{new}} < tolerance$$

For the creation of the master files used in this project, a tolerance of 5 was used for all the files. This process, with no other inputs, was used in the dark and bias images, but for the flats a scaling factor of $\frac{1}{m}$, where m is the median of the distribution was introduced. The reason behind this, as can be seen in Figure 5, is because there is some difference between the mean and median values and using them without a scaling factor would cause the pixels with higher counts to have an increased weight. Again, we use the median (or in this case the inverse median) because it is not as affected by extreme values as the average.

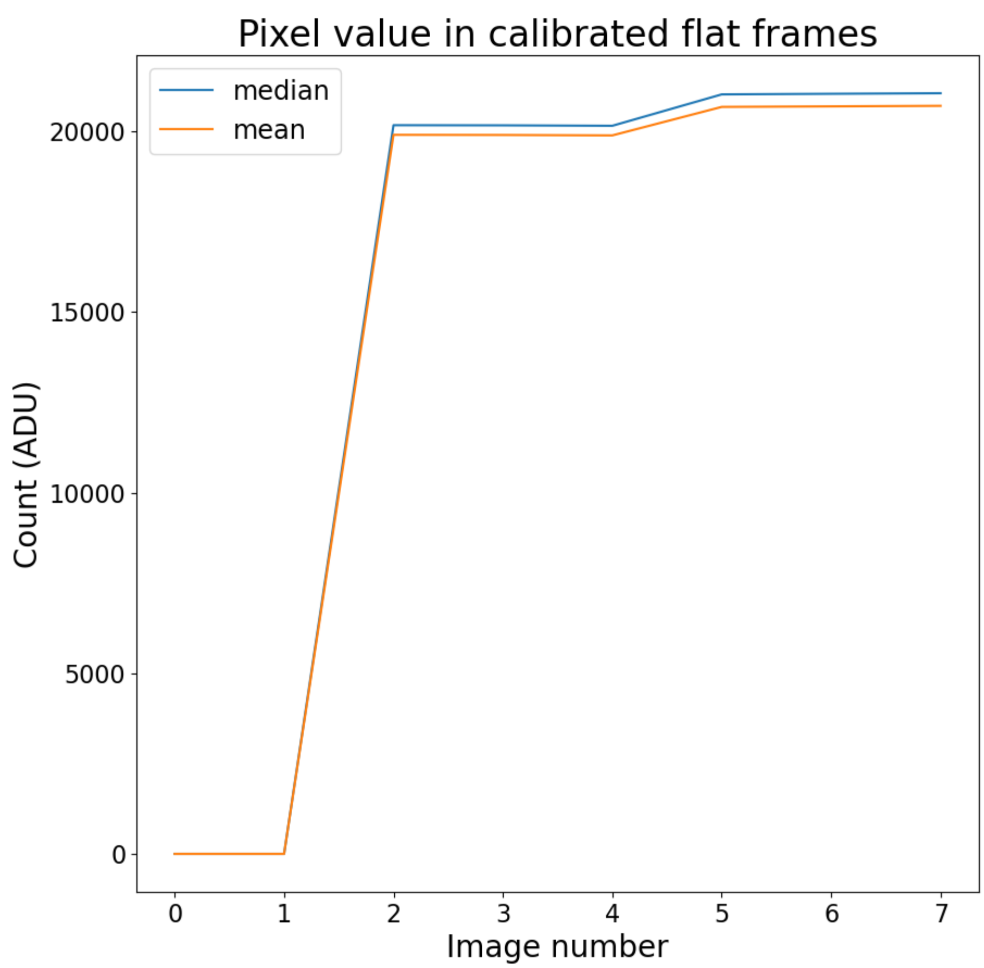

Figure 5: Pixel value in flat frames. The mean and the median are not always the same, and as such a scaling factor in the form of the inverse median is used.

The image can be seen before and after scaling in Figure 6. The main difference is in the color bar itself. In this case, the effect of scaling is not enormous, and the results are similar if the factor is altogether removed, but it may be useful in other datasets, and since going over each flat individually and deciding whether it needs to be scaled is a tedious process, it is left in for the purposes of this investigation.

(a) Before scaling

(b) After scaling

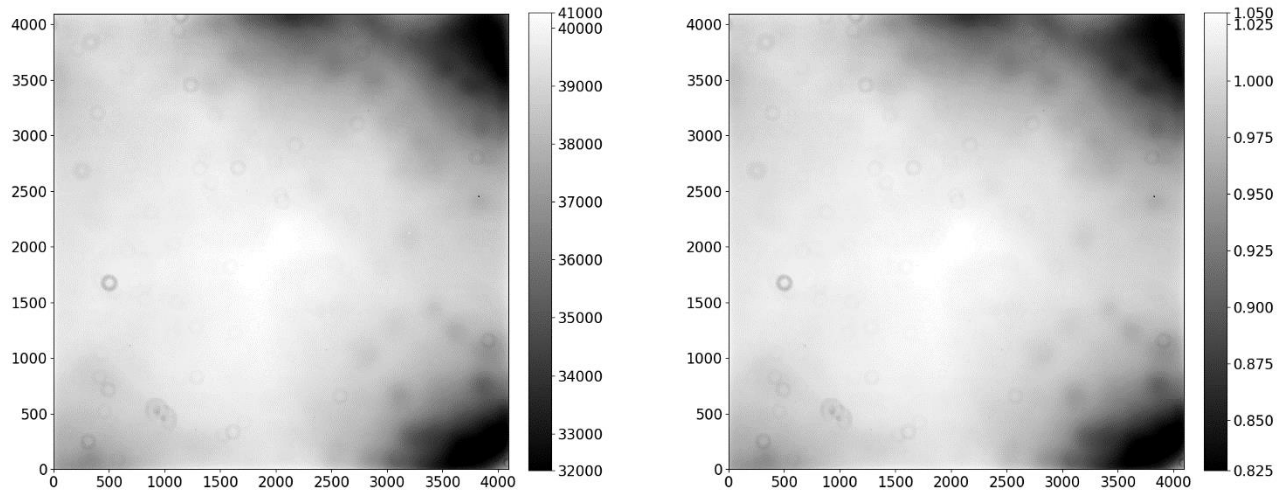

Figure 6: Effect of scaling on the master flat. The main difference lies in the color bar and the value of the pixels.