Non-iterative blind deconvolution algorithm and its improvement for astronomical imaging

Memo: Image Restoration Workflow

Maxim Usatov
Černochova 1291/4, Košíře, Prague, Czech Republic

January 10, 2011

This memo provides guidelines on the workflow for image restoration using non-iterative blind deconvolution algorithm. There are at least three variables in the blind algorithm that are not known a priori – the filter size, noise estimation and compression ratio. The workflow suggests a convenient procedure for exploring all the solution space.

Introduction

Because the image restoration problem is highly contingent to individual taste preferences and perception, the task pursued and displaying equipment, there may be multiple solutions available for a single input image. It is, thus, viable to explore all the solutions in order to select the best final result. The workflow I suggest can be graphically represented in the chart below:

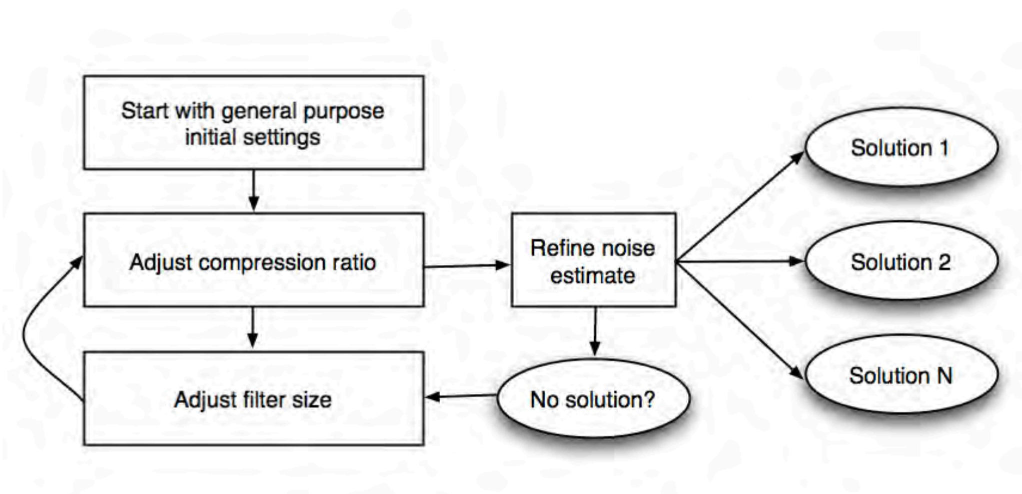

Fig 1. Suggested workflow chart.

Start

We first define the so called general purpose settings to start with. Those are experimentally set the way to provide a fair solution for an average image in a class of images one is usually working with. The initial solution derived under these settings does not have to be perfect. At this point we are mostly interested to get a positive image restoration dynamics. For lunar and planetary images, for example, I use a Gaussian weighed smoothing filter, size of 25 pixels, compression ratio of 0.75 and noise estimate (NE) of 0.5. Please notice that for the sake of readability we use NE multiplied by 1,000 in this paper, i.e. NE of 0.5 equals N=0.0005 actually used on the Wiener filtering step of the deconvolution algorithm. The same settings may work well for other images. Because NE usually varies highly from image to image, it is worth to take an initial guess of it initally, before making any further adjustments. Exaggerated example of the initial NE estimation is shown in figures below. It is important not to set NE too high in order to make sure the restoration process is not suppressed completely.

Fig 2. Alpine Valley Rille on the Moon. Image obtained by Maxim Usatov with 0.25m Maksutov-Cassegrain telescope in Prague, Czech Republic in poor seeing conditions.

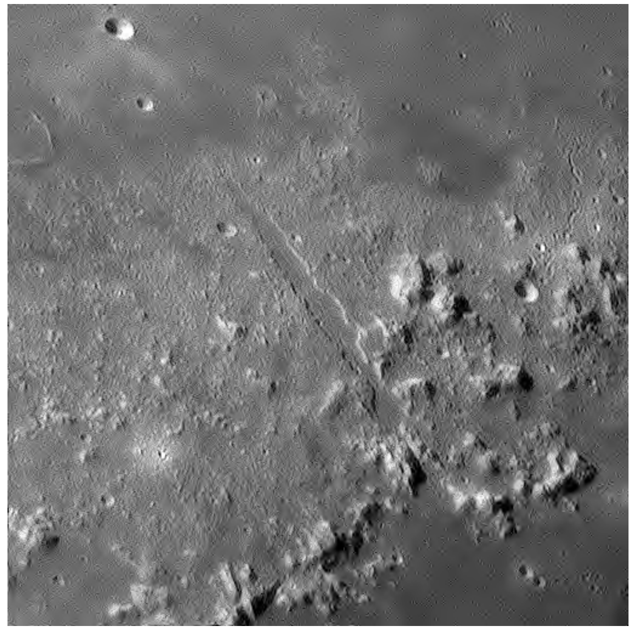

Fig 3. Start with general purpose settings: filter size 25 pixels, compression ratio 0.75, NE=0.1, results in sharp image with too much high-frequency noise.

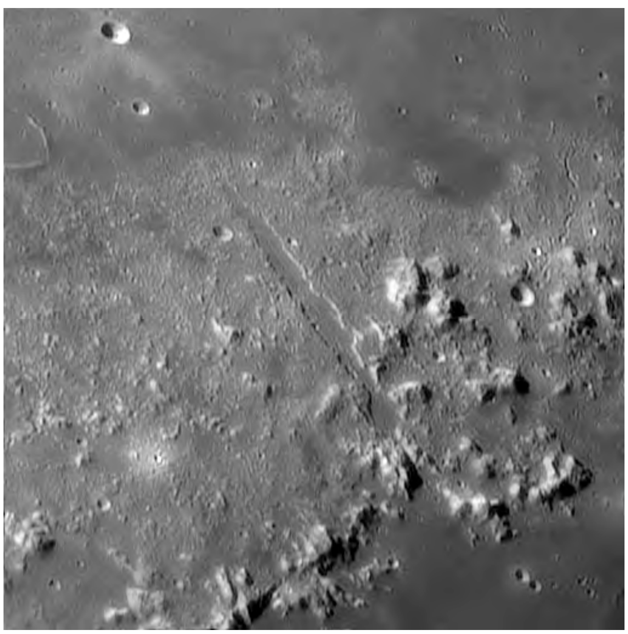

Fig 4. Same settings as with figure 3, however with NE=1.5 results in too blurry image, high frequency features are suppressed.

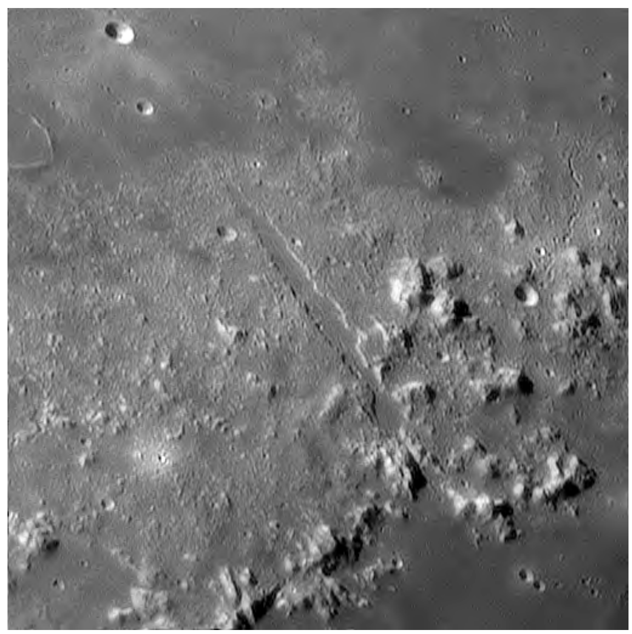

Fig 5. Same settings as with figure 3, however with NE=0.5 results in about right balance for the start. Although this image is not a perfect solution, positive dynamics has been established at this step.

Adjusting Compression Ratio

The next step would be to adjust the compression ratio and ultimately find a good restoration solution, with the filter size remaining constant. Because compression ratio varies in the [0, 1] range, it is relatively easy to find a solution if such indeed exists for a given filter size. We may start adjusting the ratio by 0.1 steps and then refining it by 0.05 or even smaller steps, if necessary. One can also get a good feel on how the deconvolved image reacts on settings changes during this adjustment. The higher the compression ratio is, the more emphasis is given to higher frequency components of an image; and vice versa, low compression ratios will enhance lower frequencies of the input image, preserving higher frequencies at their original levels. The goal for this compression ratio adjustment is to find the correct compression curve that focuses the restoration process on important features of the image. In the Moon's Alpine Valley Rille example used, compression ratio set too high will result in the restoration process 'overshooting' important features such as craters and mountains, which can be generalized as medium frequency features. The image will appear flat and featureless, as if the image is not restored, rather degraded further. In addition, such a high compression ratio will amplify high frequency additive noise. While it is possible to suppress such noise by increasing the NE, it is possible to identify the ratio being right or wrong by assessing the impact of the restoration process on individual image features. Examples shown in figures below.

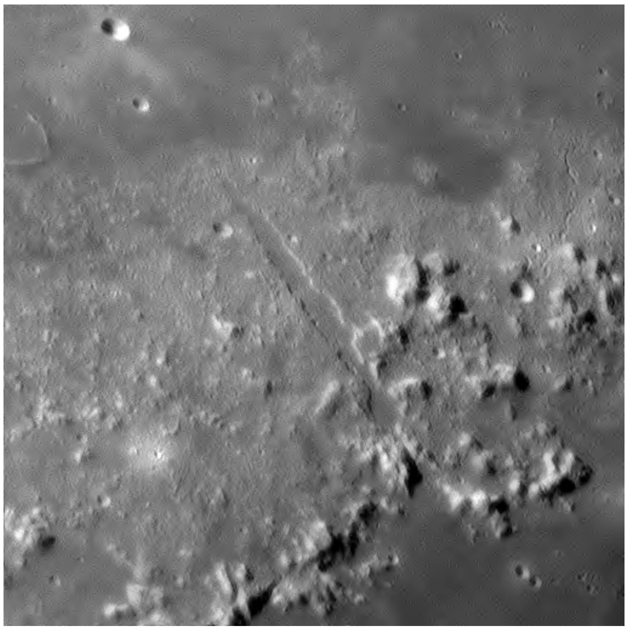

Fig 6. Filter size 25 pixels, NE=0.5 and compression ratio of 0.9 results in restoration process focusing on very high frequency components which are not present in the image. The result of this is amplified high frequency noise and no restoration for important image features.

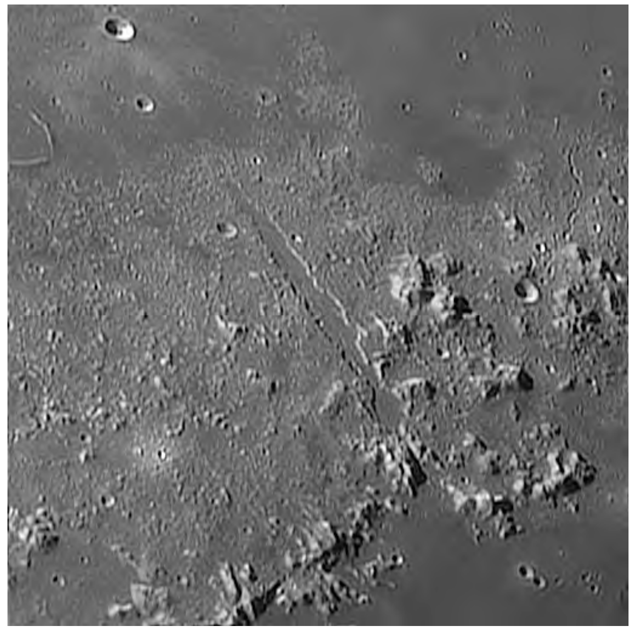

Fig 7. Filter size 25 pixels, NE=0.5 and compression ratio of 0.1 results in low frequency boost that creates unnatural look and lack of sharpness.

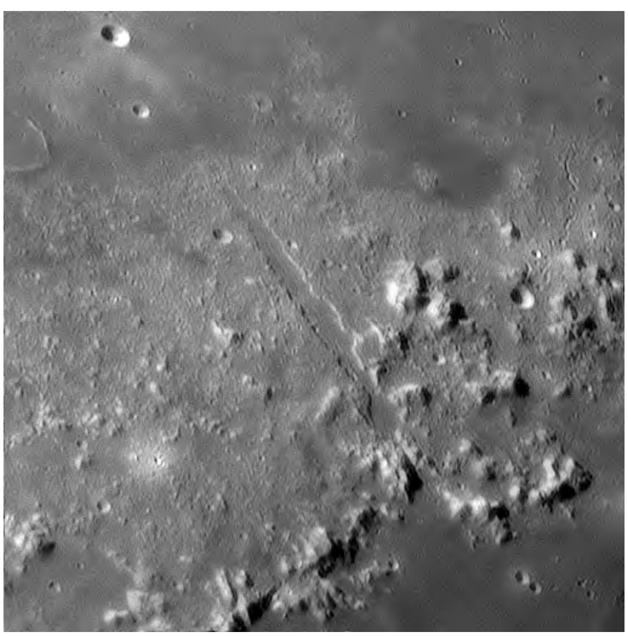

Fig 8. Filter size 25 pixels, NE=0.5 and compression ratio of 0.82 is a good solution as important spectrum of features is restored.

Refining Noise Estimate

Once good solution is found on the previous step, it makes sense to refine the NE a little bit further to suppress additive noise or to allow more restoration at higher frequency ranges, increasing the overall sharpness of the restored image. A good test bed for NE adjustments would be featureless, uniform regions of an image where such noise is visible at most. Once NE is fine-tuned, the final image is saved as one of the possible solutions. One may finish at this step or explore other solutions available.

No Solution Found?

In our Moon example, the initial filter size of 25 pixels turned out to be a rather good estimate. In some cases it will not be possible to find a good combination of NE and compression ratio for a guessed filter size, i.e. all solutions are going to appear wrong. This is especially true if the initial general purpose settings were inappropriate for the input image one is trying to restore. Following the workflow chart, if no solution is found one has to adjust the filter size and continue from the compression ratio adjustment step. Because filter size can vary from a few to a few hundred pixels, it is the most difficult unknown to guess. If the result appears wrong or one can't establish positive restoration dynamics, it makes sense to adjust the filter size in broad steps – by 10 or more pixels, and then refining it with 1-2 pixel steps.

Exploring Other Solutions

Once a good solution is found, the next step would be to adjust the filter size and search for alternative solutions for the same problem, re-starting from the compression ratio adjustment step. Generally, if the filter size is decreased, the compression ratio will have to be increased, and vice versa. Many solutions may look close to each other because both filter size and compression ratio seem to compensate the effects of each other. However, an ideal filter size is such that rapidly varying image signal is smoothed while slowly varying degradation data is preserved. This allows for the deconvolution algorithm to build a correct blurring function that is then used to restore the image. Too small filter will result in rapidly varying image signal to remain; this leads to the deconvolution process attempting to account that image signal as the degradation, resulting in artifacts or even corrupted image. Too large smoothing filter will filter out the broad degradation part from the input, thus creating weak blurring function leading to the lack of restoration. Exaggerated cases are shown in figures below. Some of the downside effects observed can be compensated by either the compression ratio or the filter size, however an image restored with an ideal filter size should be the best compromise between sharpness and noise level. Because it is possible to identify filter size being too small or too large, the viable filter size range is limited. This limitation allows to explore all the available solutions by adjusting the compression ratio per each filter size iteration.

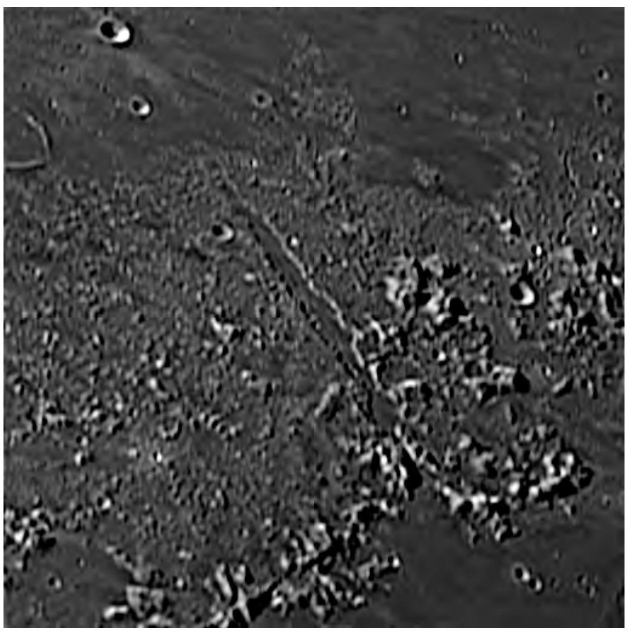

Fig 9. Filter size 5 pixels, NE=0.5 and compression ratio of 0.82 – too small filter results in algorithm erroneously using true image signal to estimate blurring function.

Fig 10. Filter size 50 pixels, NE=0.5 and compression ratio of 0.82 – too large filter results in no information left about degradation, weak blurring function and lack of restoration.

Using Improved Blind Algrorithm

Exactly the same workflow can be used with improved version of the algorithm. In addion to the unknown filter size, NE and compression ratio, one will have to take into account the unknown Fried's r_0 parameter. All other parameters related to the estimation of atmospheric MTF are known a priori and should be set fixed at the start.

One way to estimate $r_0$would be to re-use settings of the best solution derived using non-improved blind algorithm. A good sign for valid r_0 estimation would be a solution almost identical to a non-improved solution, with the same settings. Once the initial r_0 is estimated, it is viable to increase the r_0 by small steps and decrease the filter size, keeping the compression ratio and NE the same. (Or increasing the compression ratio.) The rationale for this is that the 'atmospheric' part of the algorithm compensates for the broad Gaussian-like atmospheric degradation, and this allows to focus the blind part of the algorithm on more narrow degradations, such as defocus or lens intrinsic aberrations.