Image Processing Project

A non-local algorithm for image denoising

Supervisor: Pro. Désiré Sidibé

MAIA 1
Umamaheswaran RAMAN KUMAR

Contents

1. Introduction

Over the years, a variety of methods have been introduced to remove noise from digital images, such as Gaussian filtering, anisotropic filtering, Total Variation minimization and the neighborhood filtering. Even though they may be very different in tools it must be emphasized that a wide class share the same basic remark: denoising is achieved by averaging. This averaging may be performed locally or in the frequency domain. However, many of these algorithms degrade or remove the fine details and texture of the image in addition to the noise. Now, the non-local means algorithm assumes that the image contains an extensive amount of redundancy. These redundancies can then be exploited to average similar patches across the whole image to remove the noise in the image. In this project, we implemented the non-local means algorithm and compared it to other denoising methods using the method noise measurement by ourselves.

2. Problem Statement

Previous methods attempt to separate the image into the smooth part (true image) and the oscillatory part (noise) by removing the higher frequencies from the lower frequencies. However, not all images are smooth. Images can contain fine details and structures which have high frequencies. When the high frequencies are removed, the high frequency content of the true image will be removed along with the high frequency noise because the methods cannot tell the difference between the noise and true image. This will result in a loss of fine detail in the denoised image. Also, nothing is done to remove the low frequency noise from the image. Low frequency noise will remain in the image even after denoising. The other problem with these filters is that comparing only grey level values in a single pixel is not so robust when these values are noisy. Because of this loss of detail, the paper authors have developed the non-local means algorithm.

3. Non-local Means Method

The non-local means algorithm assumes the image contains an extensive amount of self-similarity. The paper authors originally developed the concept of self-similarity for texture synthesis. An example of self-similarity is displayed in Figure 1. The figure shows three pixels *p*, *q1*, and *q2* and their respective neighborhoods. The neighborhoods of pixels *p* and *q1* are similar, but the neighborhoods of pixels *p* and *q2* are not similar. Adjacent pixels tend to have similar neighborhoods, but non-adjacent pixels will also have similar neighborhoods when there is structure in the image. For example, in Figure 1, most of the pixels in the same column as *p* will have similar neighborhoods to *p*'s neighborhood. The self-similarity assumption can be exploited to denoise an image. Pixels with similar neighborhoods can be used to determine the denoised value of a pixel.

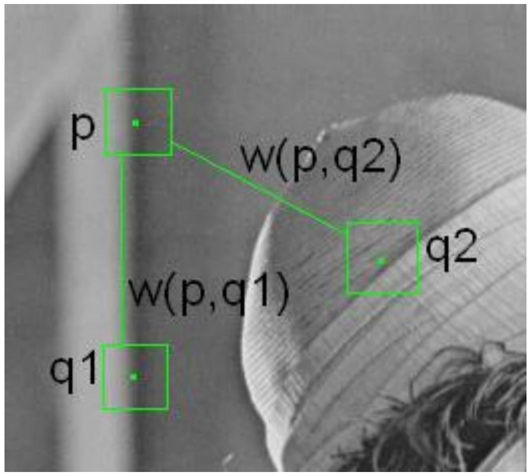


Figure 1: Example of self-similarity in an image. Pixels *p* and *q1* have similar neighborhoods, but pixels *p* and *q2* do not have similar neighborhoods. Because of this, pixel *q1* will have a stronger influence on the denoised value of *p* than *q2*.

Each pixel *p* of the non-local means denoised image is computed with the following formula:

$$NL(V)(p)=\sum_{q\in V} w(p,q)V(q)$$

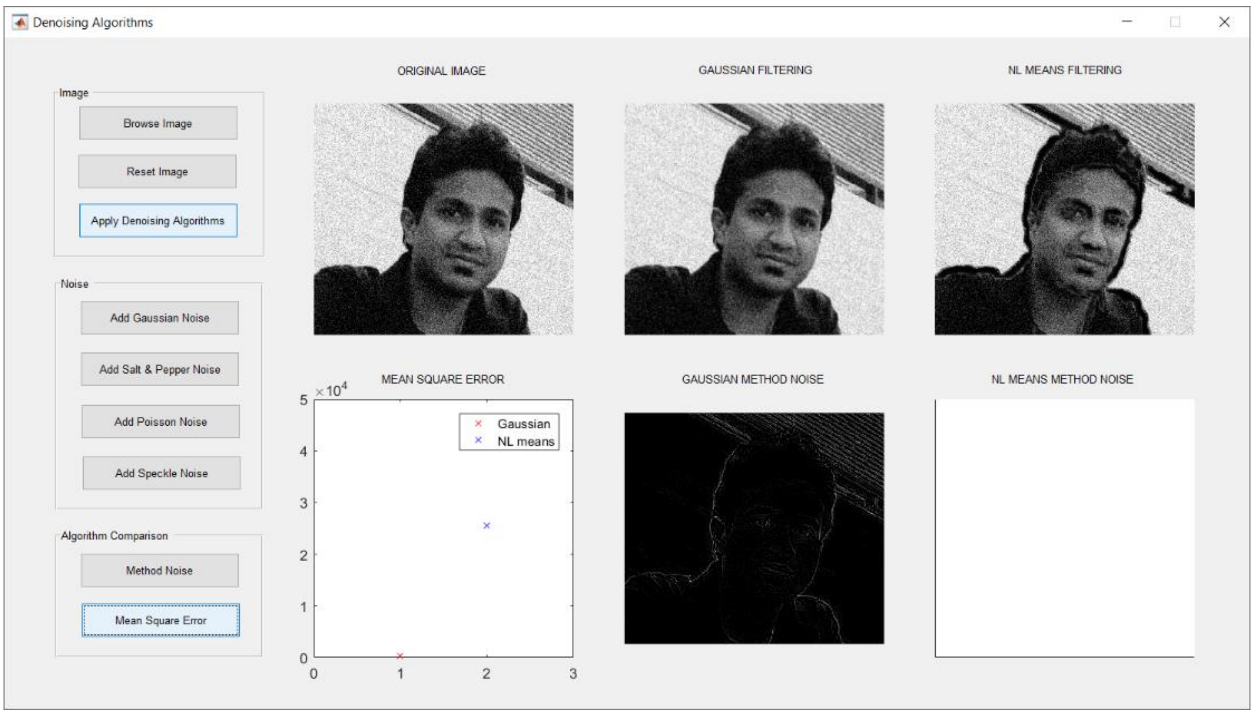
where *V* is the noisy image, and weights *w(p,q)* meet the following conditions $0\leq w(p,q)\leq 1$ and $\sum_q w(p,q)=1$. Each pixel is a weighted average of all the pixels in the image. The weights are based on the similarity between the neighborhoods of pixels *p* and *q*. For example, in Figure1 above the weight *w(p,q1)* is much greater than *w(p,q2)* because pixels *p* and *q1* have similar neighborhoods and pixels *p* and *q2* do not have similar neighborhoods. In order to compute the similarity, a neighborhood must be defined. Let N_i be the square neighborhood centered about pixel *i* with a user-defined radius *Rsim*. To compute the similarity between two neighborhoods take the weighted sum of squares difference between the two neighborhoods. *F* is the neighborhood filter applied to the squared difference of the neighborhoods. The weights can then be computed using the following formula:

$$w(i,j) = \frac{1}{Z(i)}\, e^{-\frac{||v(N_i)-v(N_j)||^2_{2,a}}{h^2}}$$

Z(i) is the normalizing constant defined as $Z(i) = \sum_j e^{-\frac{||v(N_i)-v(N_j)||^2_{2,a}}{h^2}}$. *h* is the weight-decay control parameter.

4. Implementation

The below figure shows the Graphical User Interface developed for the project.



In the project, we have implemented non-local means filtering and compared with Gaussian filter. These methods will be compared using two different criteria proposed in the paper.

1) Mean Square Error (MSE)

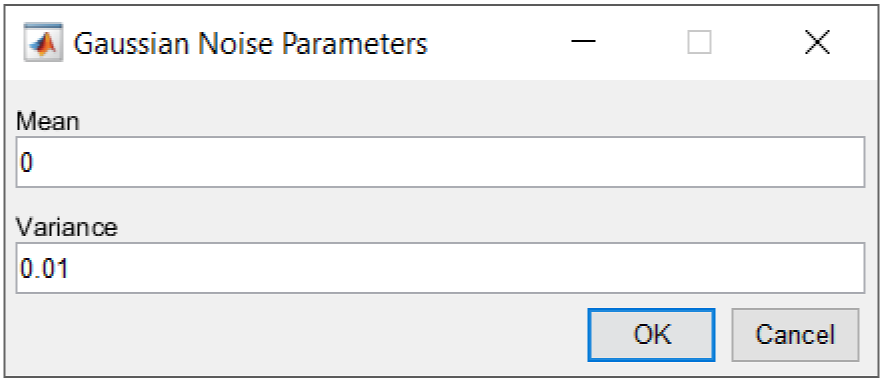
   The difference between the original image and the denoise image squared

2) Method Noise

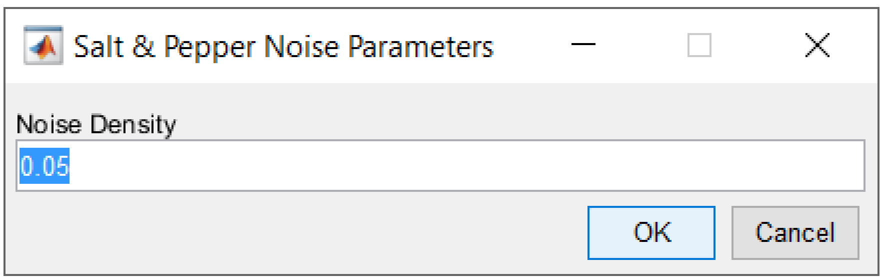
   The difference between the original and the filter applied on the original image

Additional functionalities are provided to add noise to the input image. Listed below are the noises that can be added with their parameters.

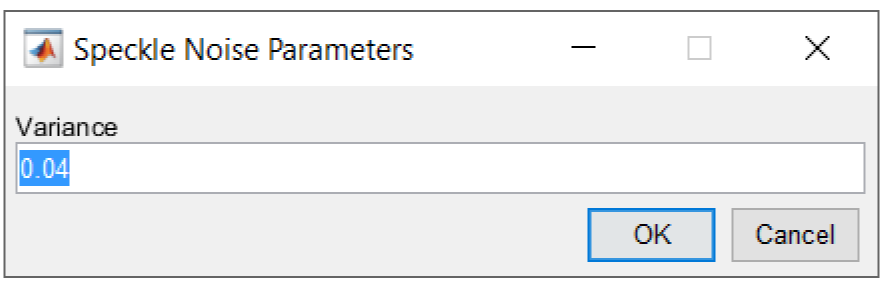
1) Gaussian noise



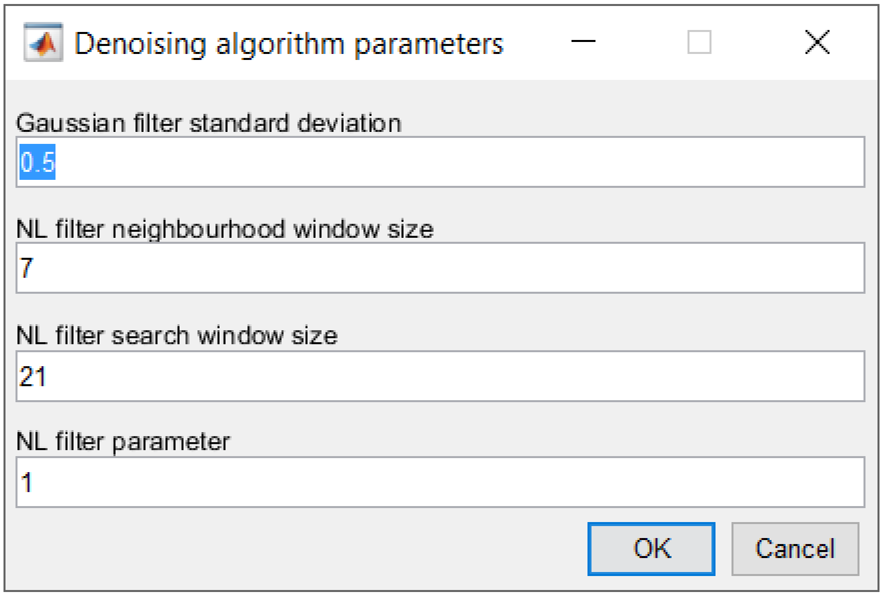
2) Salt & Pepper noise



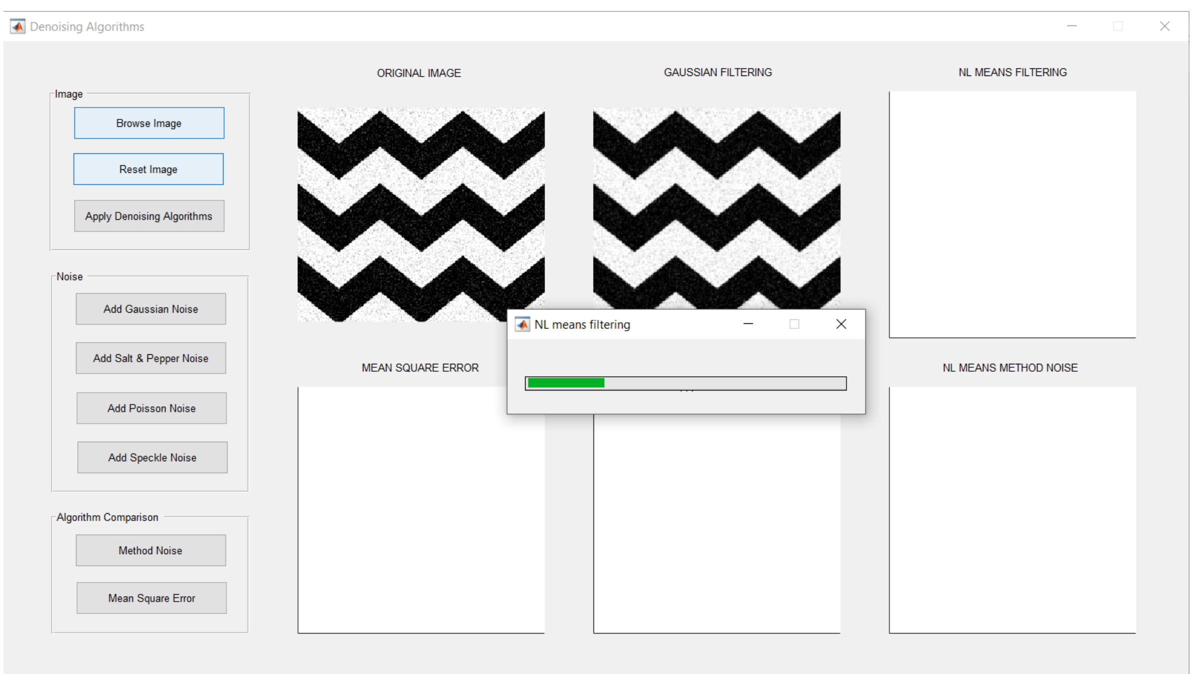
3) Poisson noise
4) Speckle noise



The below parameters can be set when we want to test the denoising parameters. The first parameter applies to the Gaussian filter and the other 3 are for the NL means filter.



It is observed that the NL means algorithm take more time for computation and hence a progress bar is added to show the progress of the filtering.



5. Conclusion

From our results, Gaussian filtering performed properly and the resulting images show small loss in detail and still contain noise. The NL means algorithm is implemented on our own and does not work as expected as specified in the paper. But the results are comparable when the filter size is considerably increased.