## **2. Algorithms**

### **2.1. 4 Sigma filter**

The 2-D 5 x 5 sigma filter takes 25 pixels as input. If we consider as the pixel of interest, the input pixels for the 5 x 5 sigma filter are

After sigma filter, the output

Here,

,

and

Overall, sigma filter averages values belonging to the same distribution as the interest input , while excluding noisy outliers in the input.

2.1.6 mean and standard deviation of the interior of the large disk region

The matrix containing the large disk is isolated manually by taking the row 50 to row 181 and column 32 to column 150 out from the original image. The left panel of Figure 1 demonstrates the area cropped by this method from the image after mean filter for 5 iterations. The right panel shows it histogram. We can see the background peaks at around 20, while the large disk peaks at around 200. I also carefully checked the brightness of the large disk as well as the line inside the disk. They are all above 100 after mean filter for five iterations, since all the dark noises are gone after mean filter application. Therefore, our method to extract the large disk is to extract the coordinates of pixels whose brightness is higher than 100 based on the image after mean filer. Then the mean and standard deviation of these selected pixels are calculated. For consistency, we use the same coordinates which we extract from the result after mean filter for the results of all the five filters.

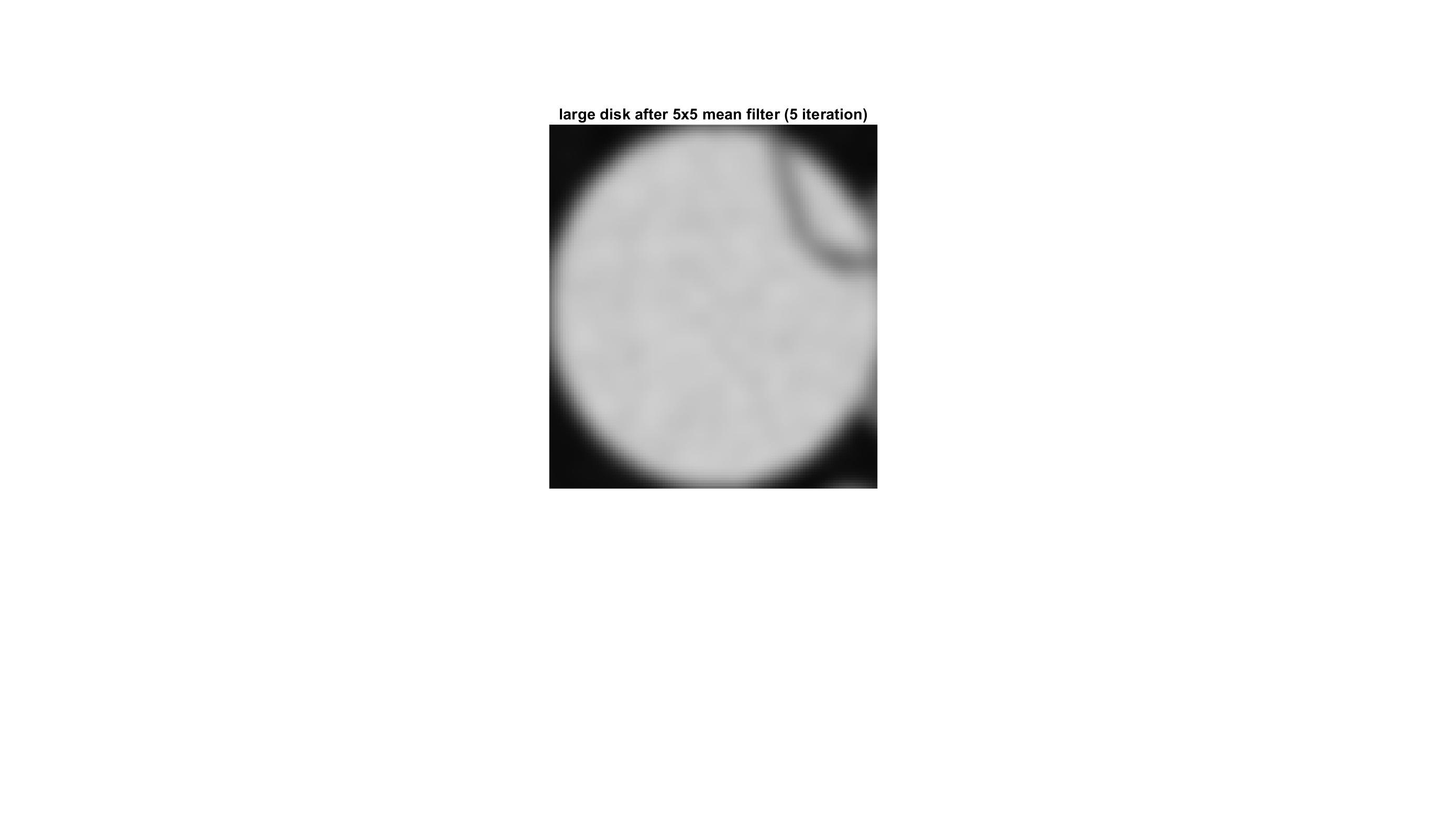
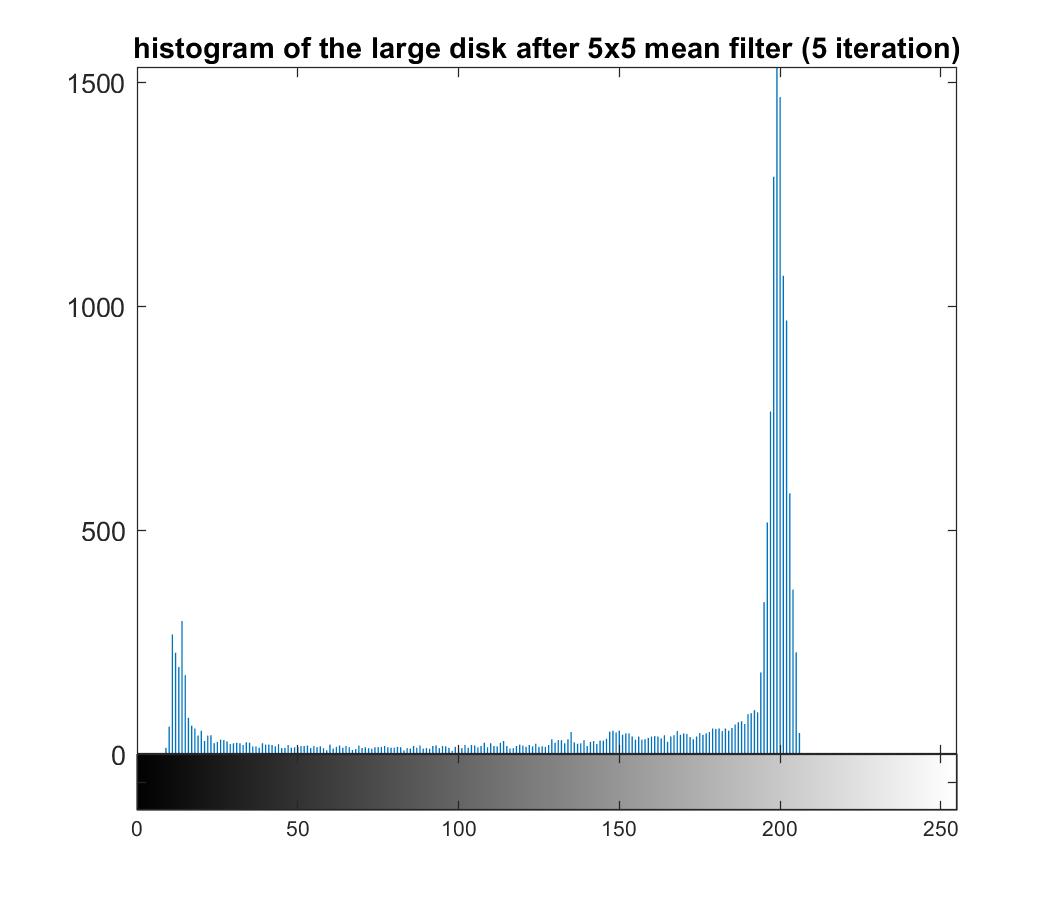
 

Figure 1 The matrix containing the large disk and its histogram after applying 5 x 5 mean filter to the original disk image.

## **3. Structure of Code**

**sigma\_filter.m**

Function sigma\_filter.m takes an image as input, and applies sigma filter to it. The output image will be the image after sigma\_filter. This function does not call any other functions.

## **C. Results**

### **1. Nonlinear Filtering**

The first part of this project focuses on applying different types of nonlinear filters and understanding their effects. 5 types of nonlinear filters have been tried to apply on the disk.gif image for up to 5 iterations. The 5 filters are: 55 mean filter, 55 median filter, 55 alpha-trimmed mean (alpha= 0.25), 55 sigma filter ( = 20), and 55 symmetric nearest-neighbor mean filter. The image output of the first iteration and fifth iteration. After 5 iteration, the gray-scale histograms were shown for each filter. Meanwhile, the mean and standard deviation of the interior of the large disk were also calculated.

Figure 2A is the original image with the gray-scale histogram. We can see the distribution is very disperse due to the presence of noise. We first applied mean filter to it. Figure 2 B showed the result of mean filter after one iteration and 5 iterations respectively. We could see mean filter sharpened the histogram greatly comparing with the original image, but the edge tended to become blurred. Mean filter takes the mean of all the 25 inputs, so it is easy to be affected by outliers. As a result, it will be easy to be affected by the background at the edge. According to the method described in 2.1.6, we extracted the interior of the large disk, and calculated the mean and standard deviation, which are 188.70 and 22.60 (Table 1).

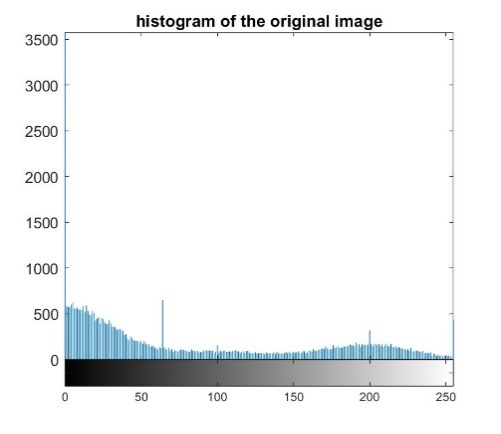
In contrast, median filter preserved the edge better than mean filter (Figure 2C left two panels). Mean filter also greatly sharpened the gray-scale histogram (Figure 2C right panel). Its mean (191.80) of the large disk is slightly higher than the result from mean filter, and its standard deviation (21.01) is slightly lower, indicating median filter is relatively less affected by the dark outlier inside the large disk.

Alpha-trimmed mean filter is also one type of mean filter, the difference is that it will eliminate the outliers. In this project, the 6 brightest and the 6 darkest pixels are not considered when we calculate the output for the interest pixel. We can see from Figure 2D, it has similar gray-scale histogram, but it preserves the edge relatively better than the mean filter. The mean (190.13) and standard deviation (22.16) of the interior of the big disk also confirmed that the alpha-trimmed mean filter is less affected by outliers which are dark points in the large disk.

Figure 2E showed the result of sigma filter, which yields sharp edges after 5 iteration. However, there are still some noises inside the disks. Sigma also take mean of the inputs, but it only calculates the mean of pixels of the same statistical distribution with the pixel of interest. Because of this operation, the background will not affect the edge, while some noises would also be kept they are too different from their surrounding pixels (larger than 40). Similar to the other three filters, sigma filter also sharpened the histogram significantly (Figure 2E). The mean brightness (192.50) of the large disk is the highest among all the five filters, the standard deviation (21.50) is also relatively low (Table 1).

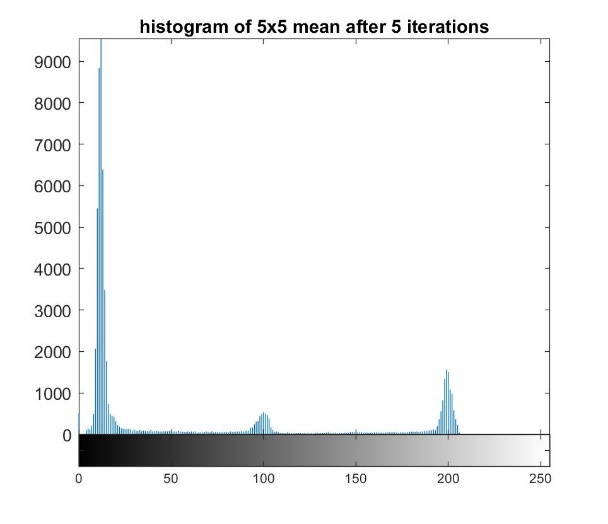
Last but not least, we applied symmetric nearest-neighbor mean filter (SNNMF). Figure 2F demonstrated that SNNMF reduces noises and preserve edges. This method first selects the most similar point to the point of interest from the two symmetrically opposite ones, then averages these selected values. This method considers similar statistical distribution, thus preserving edges. Moreover, for outliers (noises) inside disks, it still selects 13 values from the 25 inputs to average. This avoids the problem of sigma filter which could keep some noises. It has a highest mean (193.63) and standard deviation (23.67) of the large disk (Table 1).

Overall, all the 5 filters sharpened the histogram greatly, meaning all of them could reduce noises. Among them, mean filter is easy to be affected by outliers and has blurring edges. Median filter had good performance in both reducing noises and preserving edges. Alpha-trimmed mean filter, sigma filter, and symmetric nearest-neighbor mean filter all try to modify the mean filter to reduce the effect of outliers. Except that sigma filter might keep some noises, they all improved the performance of mean filter in terms of edge preserving without affecting the performance of noise reduction much.



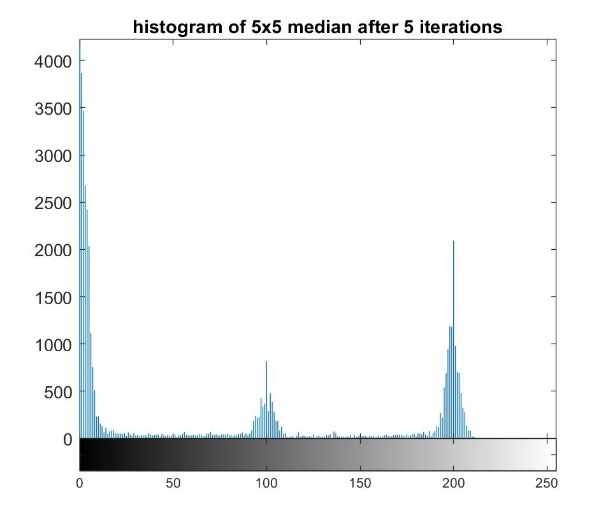
Original Image

A



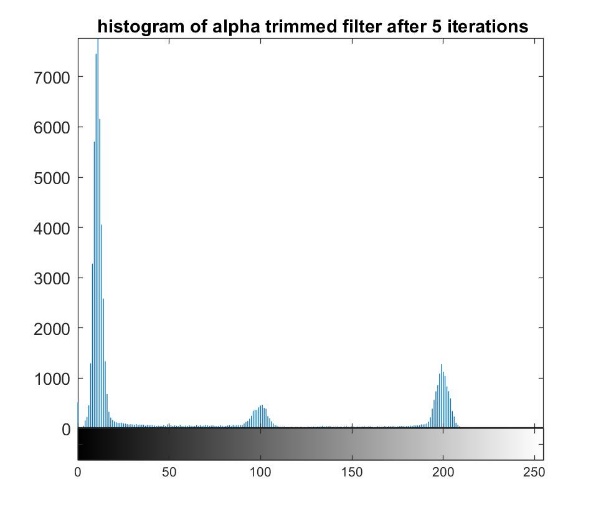
B

Mean Filter



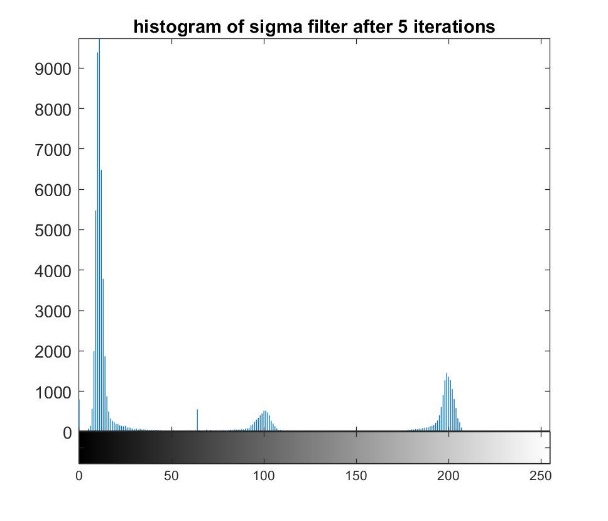
C

Median Filter



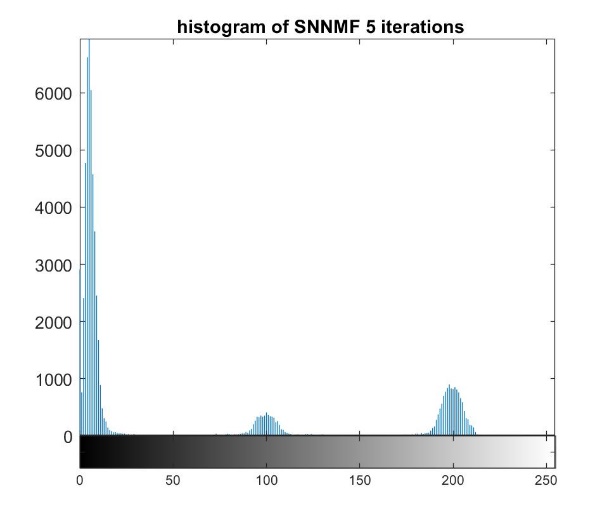
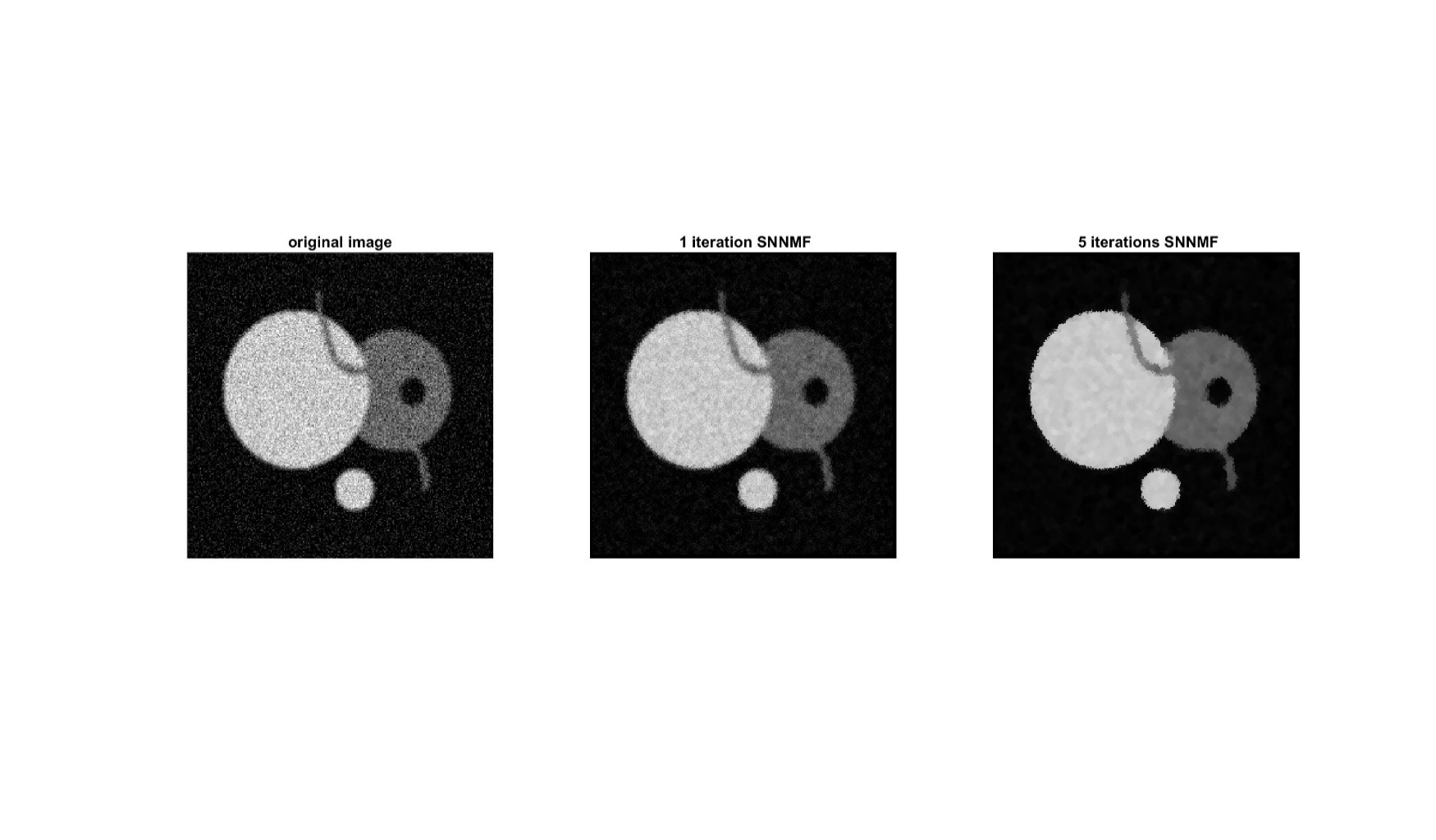
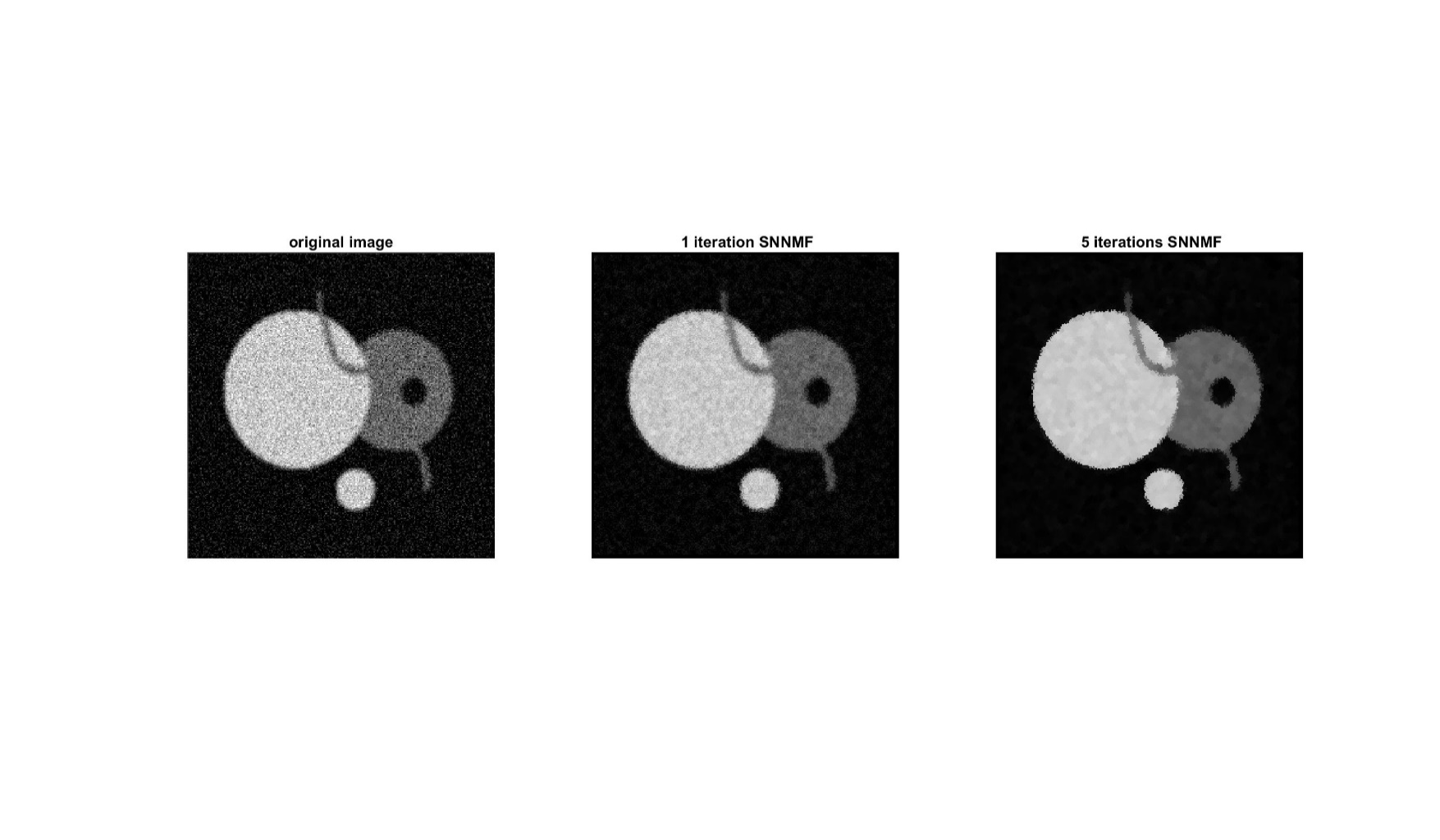
D

Alpha-Trimmed Mean Filter



Sigma Filter

E



SNNMF

F

Figure 2 The image output and histogram of the original image (A), and images after applying mean filter (B), median filter (C), alpha-trimmed mean filter (D), sigma filter (E), and symmetric nearest-neighbor mean filter (F) for 1 time and 5 times.

|  |  |  |
| --- | --- | --- |
|  | Mean | Std |
| 5x5 mean filter | 188.70 | 22.60 |
| 5x5 median filter | 191.80 | 21.02 |
| 5x5 alpha-trimmed mean filter | 190.13 | 22.16 |
| 5x5 sigma filter | 192.50 | 21.50 |
| 5x5 symmetric nearest-neighbor mean filter | 193.63 | 23.67 |

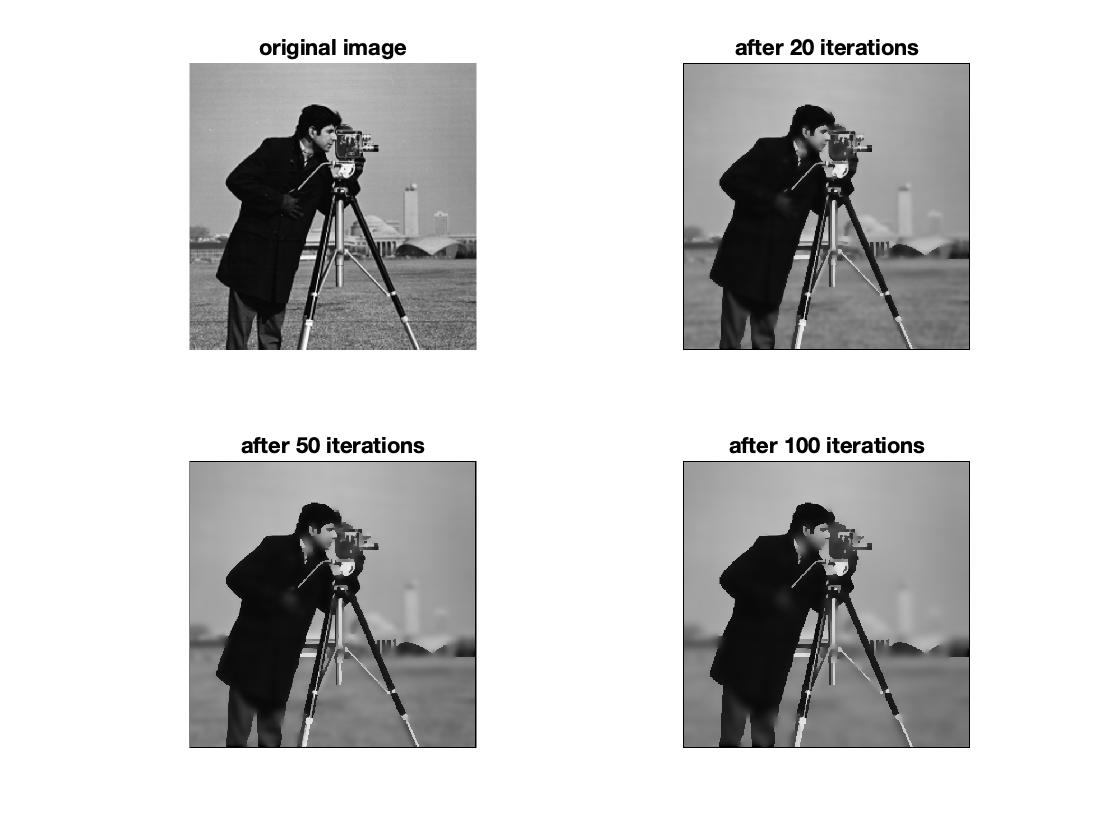
Table 1 mean and standard deviation of the interior of the large disk region for each filter after 5 iterations.

**2. Results of Anisotropic Diffusion for Image Filtering:**

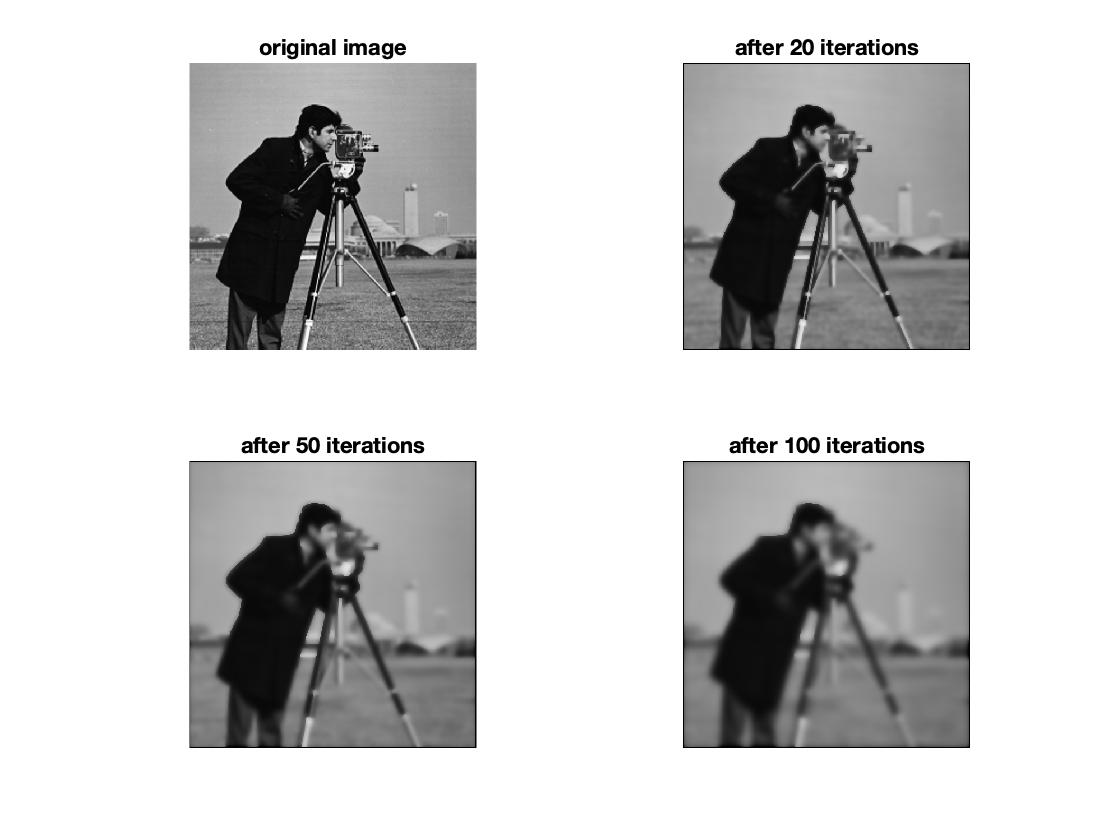
**2.1** **Anisotropic Diffusion Filter on “cwheelnoise” Image**

### **2.2** **Anisotropic Diffusion Filter on “cameraman” Image**

We processed the “cameraman” image with the same filters used above. Figure 3 showed that when k = 25, g(.) is the exponential function, each area tends to be smoothed by the anisotropic diffusion filter, but the edge or the segmentation was preserved very well. However, when g(.) is inverse quadratic function, the whole image was smoothed and the edge became very blurring (Figure 4).



*Figure 3 Applying anisotropic diffusion for up to 100 iteration. k = 25, g(.) = exponential.*



*Figure 4 Applying anisotropic diffusion for up to 100 iteration. k = 25, g(.) = inverse quadratic.*

When k is 50, the result using exponential function as the conduction coefficient function smoothened a bigger area comparing with that when k = 25. Only the edge that separates regions having higher contrast was preserved after more than 50 iterations. When the pixel value difference between two adjacent regions was relatively small, the edge was blurred as well (Figure 5). The process that uses g(.) = inverse quadratic function (Figure 6) blurred the edge even faster than k = 25 (Figure 4).

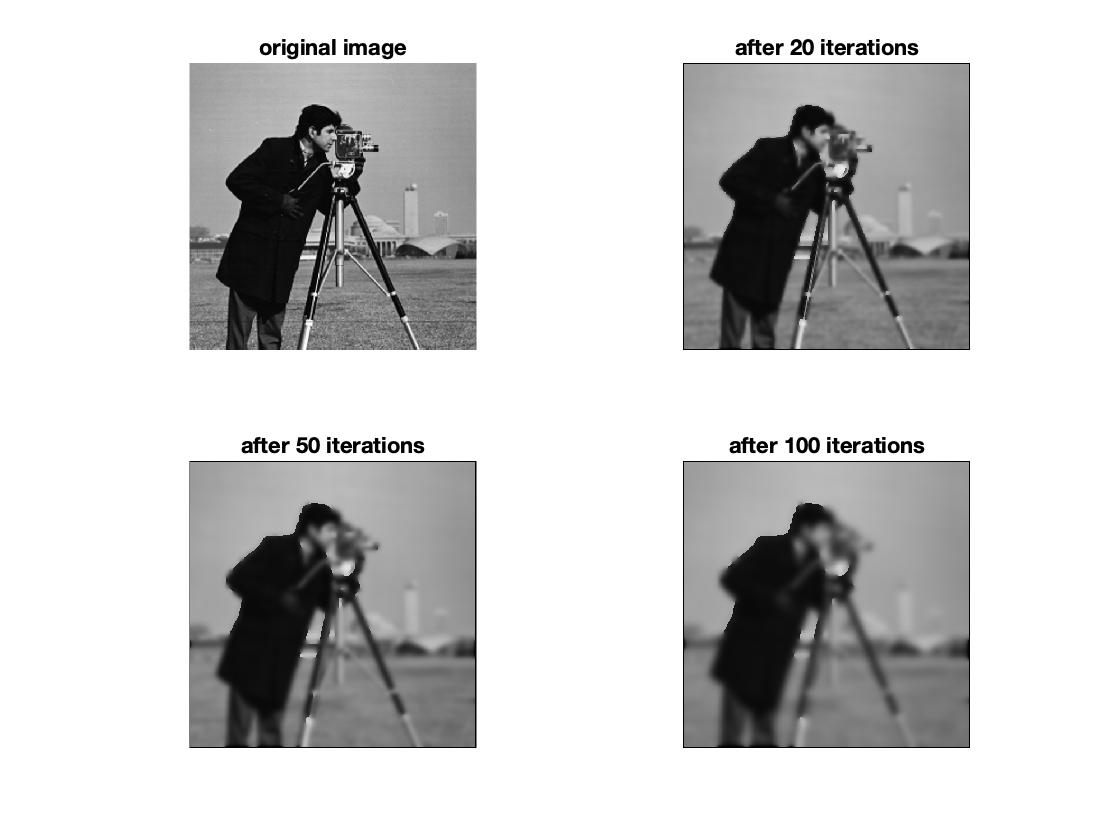


Figure 5 Applying anisotropic diffusion for up to 100 iteration. k = 50, g(.) = exponential.

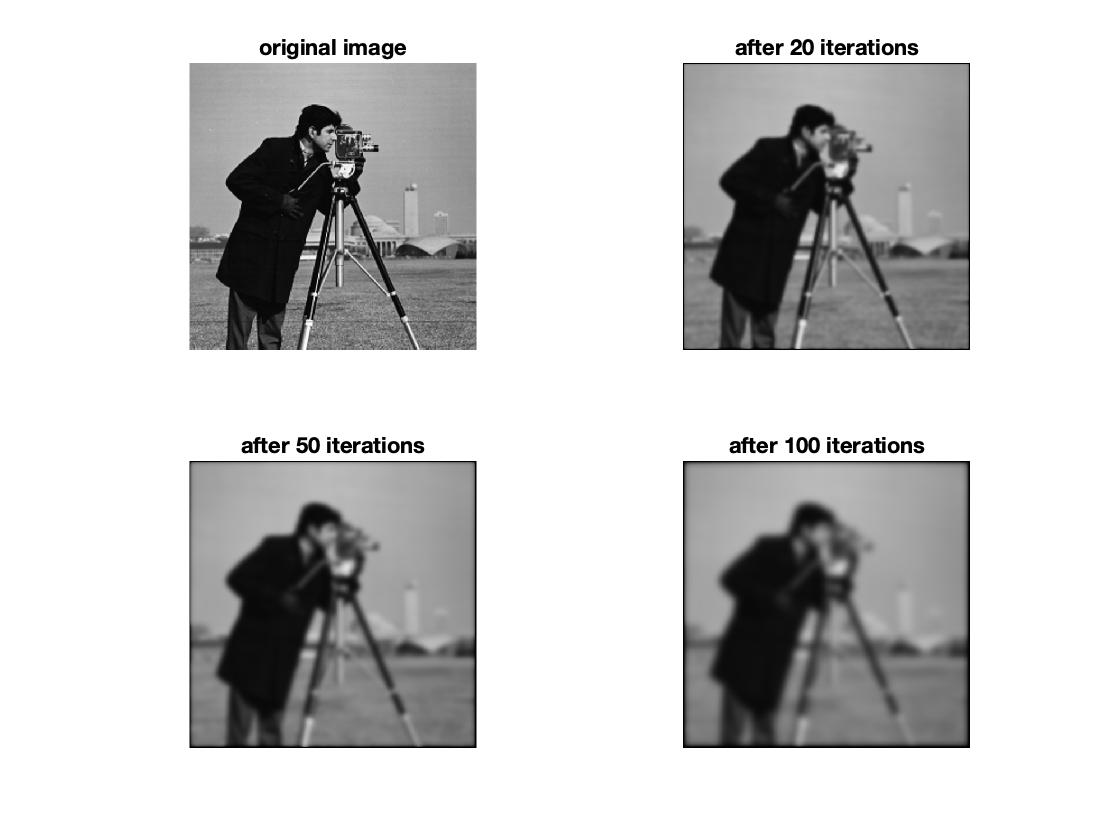


Figure 6 Applying anisotropic diffusion for up to 100 iteration. k = 50, g(.) = inverse quadratic.

**2.3 Discussion**

In general, the conduction coefficient function distinguishes the interior region and the border, and the parameter determines a threshold to distinguish the border. When k is the same, the exponential function converges faster than the inverse quadratic function, so it is more sensitive to the border, thus preserving the edges better. The inverse quadratic function, in contrast, can remove the noise in the wide region better.

The “cwheelnoise” image is relatively simple. Each region has regular and well-defined shape, and the contrast between each region is relatively high. The content in “cameraman” is relatively complex. For example, the tripod has thin and long shape, and the lower part between the grass and the lower part of the tripod is very small. As a result, when k is 25, the edges are well-preserved in both images, though the noise are not totally gone in the “cwheelnoise” image. When k becomes 50, the edges in “cwheelnoise” image are still roughly kept, but most edges are blurred in the “cameraman” image.