**CSE585/EE555:  Digital Image Processing II**

**Computer Project # 3:**

**Nonlinear Filtering and Anisotropic Diffusion**

**Yanxi Yang, Jiuchao Yin, Hongjie Liu**

**Date: 03/26/2020**

1. **Objectives**

* Study the algorithms and design different filters with given neighborhood and implement expecting applications: mean, median, alpha-trim, sigma, symmetric nearest-neighbor mean, and anisotropic diffusion.
* Get familiar with gray-scale image histograms and extract useful information from the histograms.
* Learn how to segment images from applying filters and read histograms by manually setting the threshold.
* Give observation of single iteration vs repeated application of filters on real-world images.
* Evaluate the performance of different algorithms with different parameters of anisotropic diffusion.

1. **Algorithms**

**1. Non-linear filtering algorithms**

1. **5x5 mean filter**

A 5x5 mean filter is to simply replace each pixel value in an image with the mean (average) of its neighboring pixels (window size is 5x5). Since the output pixel is linear combinations of the neighboring input pixels, thus, 5x5 mean filter is a linear filter.

= mean(…….)

where, middle element is the kth element, total number of pixels are 2N+1.

1. **5x5 median filter**

A 5x5 median filter is a non-linear filter. At each position of window, the sample values inside are ranked according to their magnitude and the middle element of this ranking is defined to be the output.[Gabouj p9]

= median (…….) [Gabouj 1.1]

where, middle element is the kth element, total number of pixels are 2N+1. In our case, k is 13 and N is 12.

1. **alpha-trimmed mean filter**

Alpha-trimmed mean filter is a subclass of order-statistical filter. The filter average a subset of samples in the filter window: the points excluded are those of very high or very low rank.[Gabouj p17]

(Gabouj 1.7)

where n= 2N+1=25, is the window length, α =0.25.

1. **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.

1. **Symmetric NNMF:**

The symmetric nearest neighbor mean filter is a 2-D filter. In this project, the size of the window is 5\*5. The function of this filter is to select the point which is the most similar to the central pixel in the window from each symmetrically opposite pair. So there are 13 pixels after the selection, and then get the average value of these 13 pixels to represent the value of the central pixel in the new image. Here is an example of symmetric nearest neighbor mean filter.

图片包含 游戏机, 灯光, 钟表, 大

描述已自动生成

Figure 1 An example of symmetric nearest neighbor mean filter.

The formula of the output with respect to Figure 1 is

The symmetric nearest neighbor mean filter can reduce the noise, sharpen edges, and reduce thin lines.

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

The implementation of anisotropic diffusion for image filtering is based on the equation

In this equation, N, S, E, W represent north, south, east, west neighbors of pixel (i, j). is a parameter and is set to be 0.25 in our project. refers to difference between pixel (i, j) and one neighbor of it, for example:

are conduction coefficients and updated after every iteration. In this project, we have tried two computation methods of it. How to compute them is shown below:

In this project, we have tried two forms of the function . The first one is the exponential formula:

The second one is the reciprocal function:

The in both functions is a parameter and the value of it is selected by us. We have tried several values of , and the different results are shown in results section.

For cwheelnoise.gif figure, the images after 0, 20, 50, 100 iterations, gray-scale histogram, plot of the line y=128 through the image, and segmented version are also shown. We select pixels which values are between 80 and 110 to be the segmented image which shows the shape of the wheel, and the range of the value is according to the gray-scale histograms, we just pick the second peak in the histogram.

The same process is also conducted on cameraman.tif image.

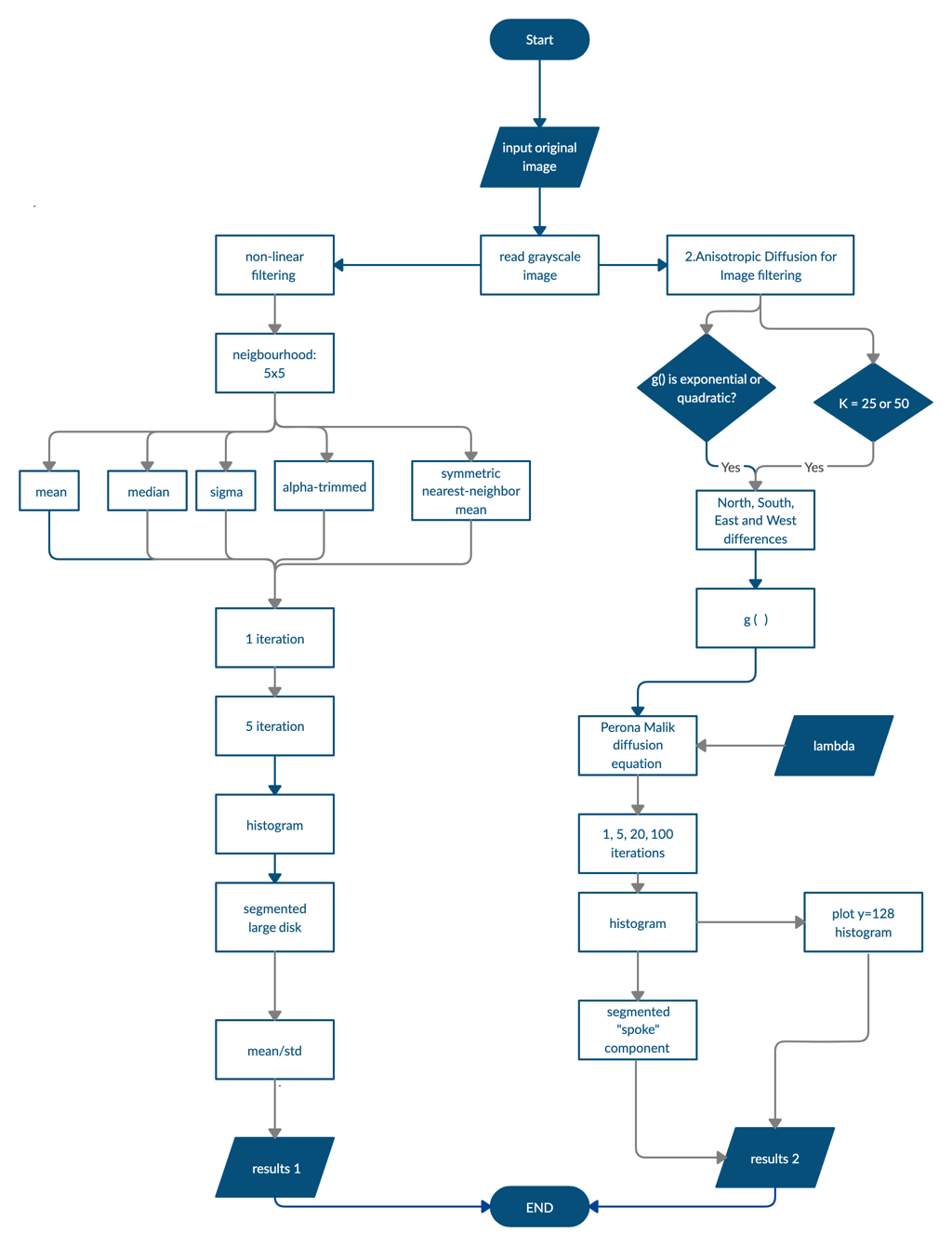
1. **Structure of codes**

**Readme:**

**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.

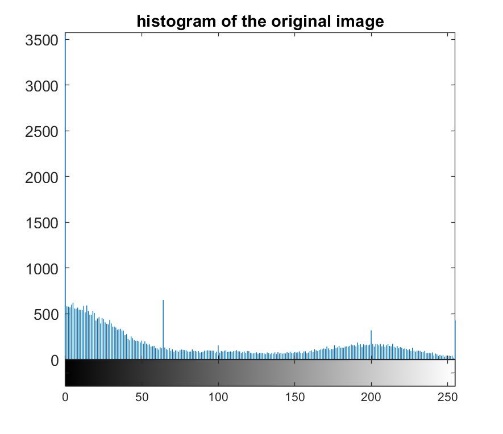
**Flow chart:**



1. **Results**

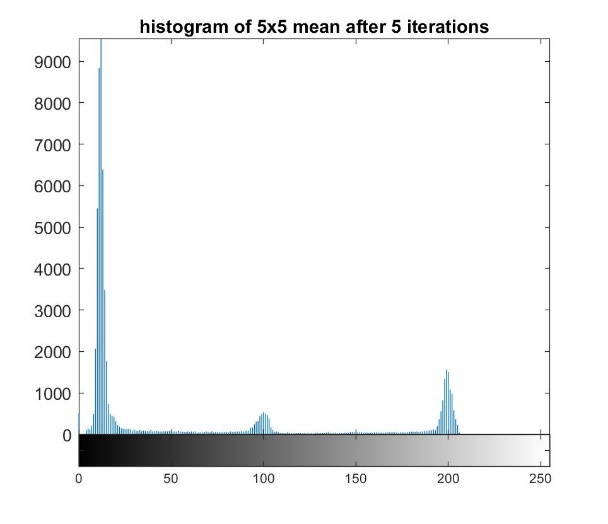
### 1. Results of Nonlinear Filtering

The first part of this project focuses on applying different types of nonlinear filters and understanding their effects.



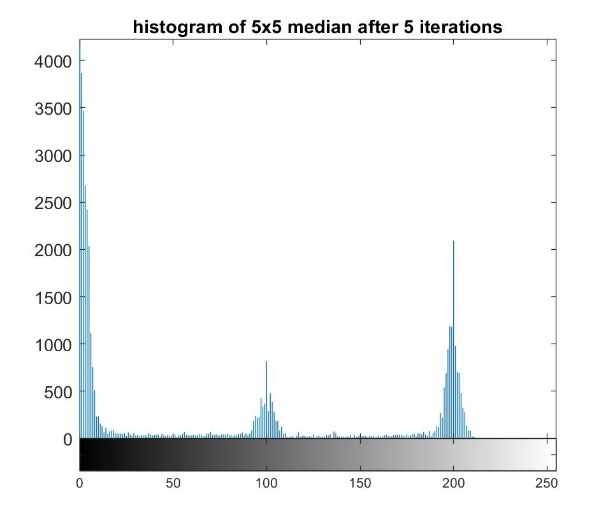
Original Image

A



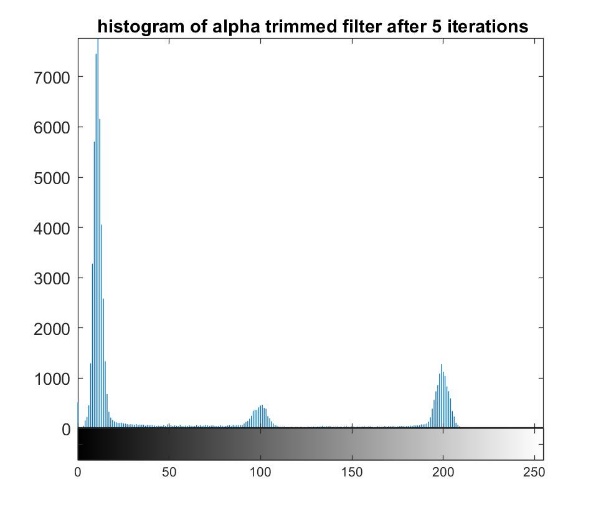
B

Mean Filter



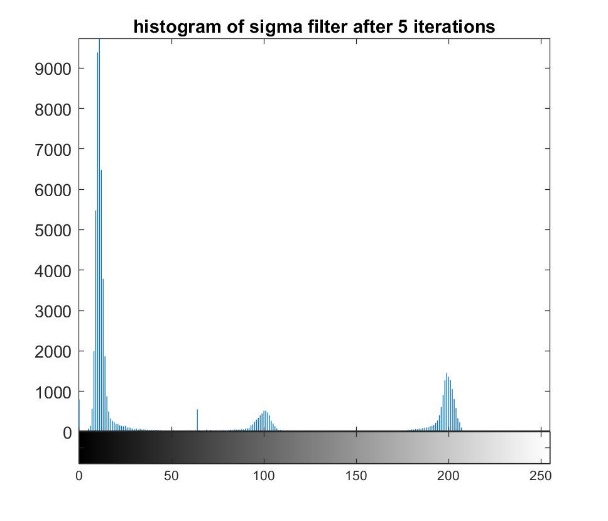
C

Median Filter



D

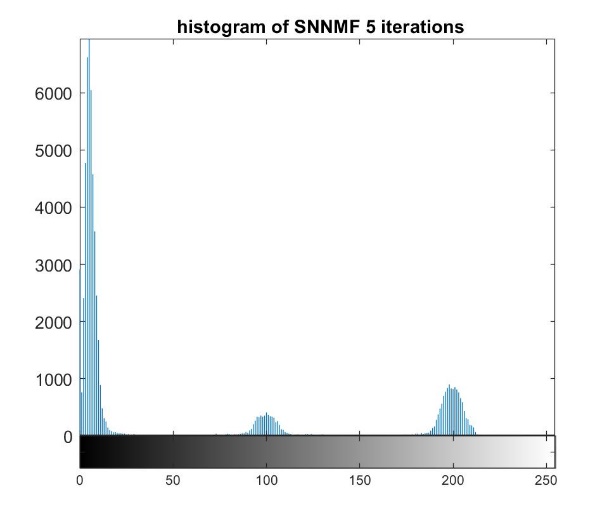
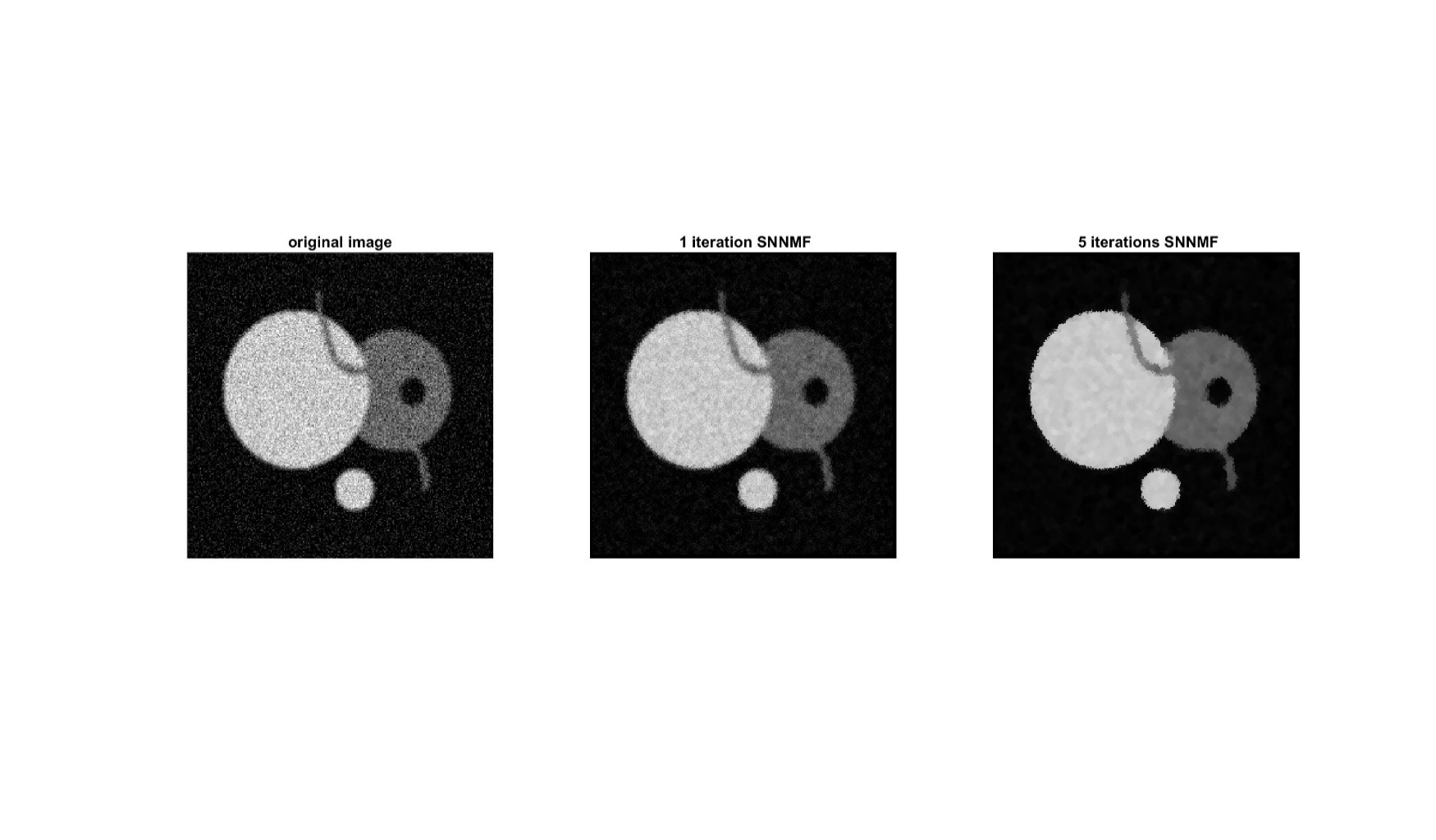
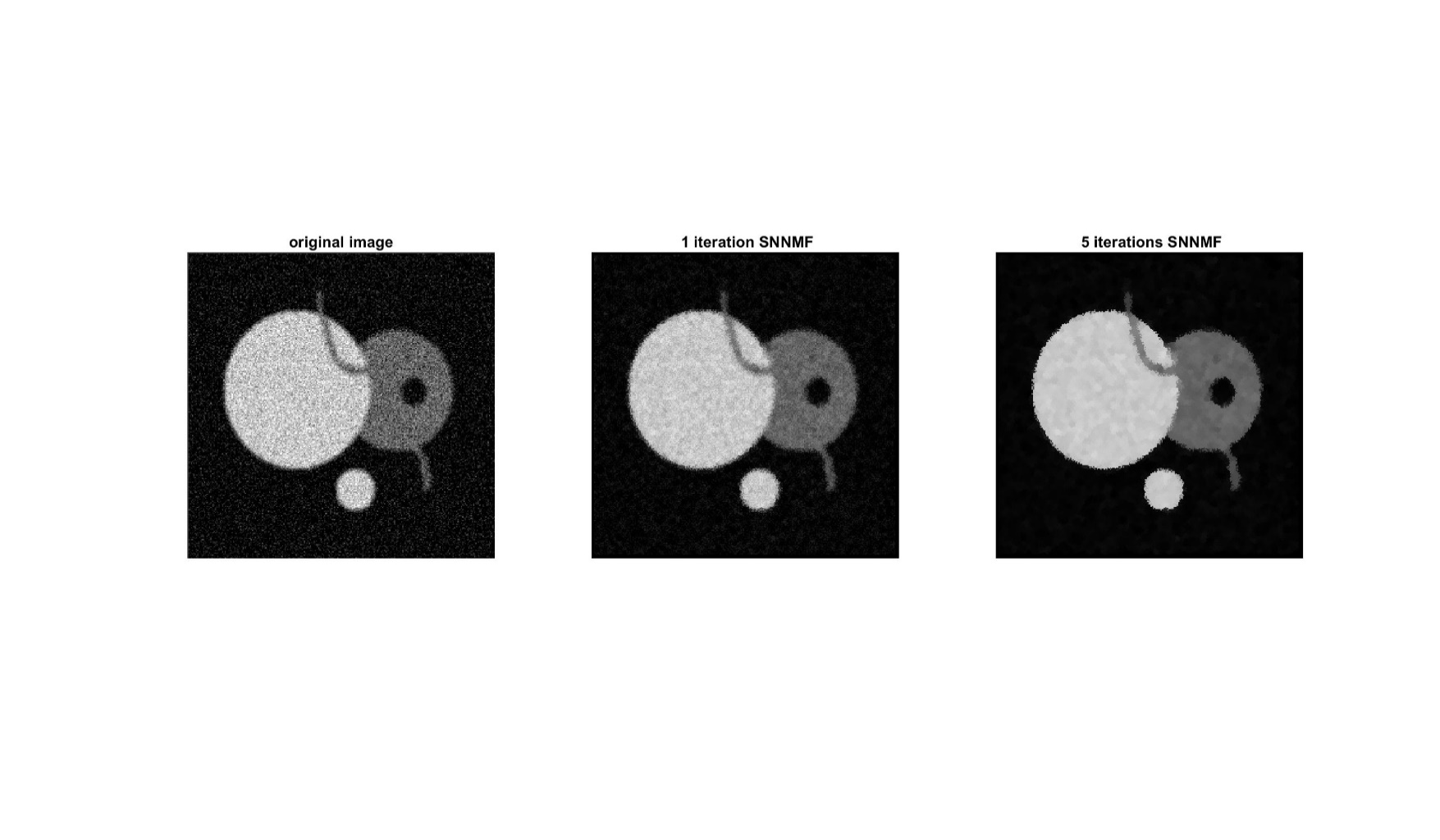
Alpha-Trimmed Mean Filter



SNNMF

Sigma Filter

E



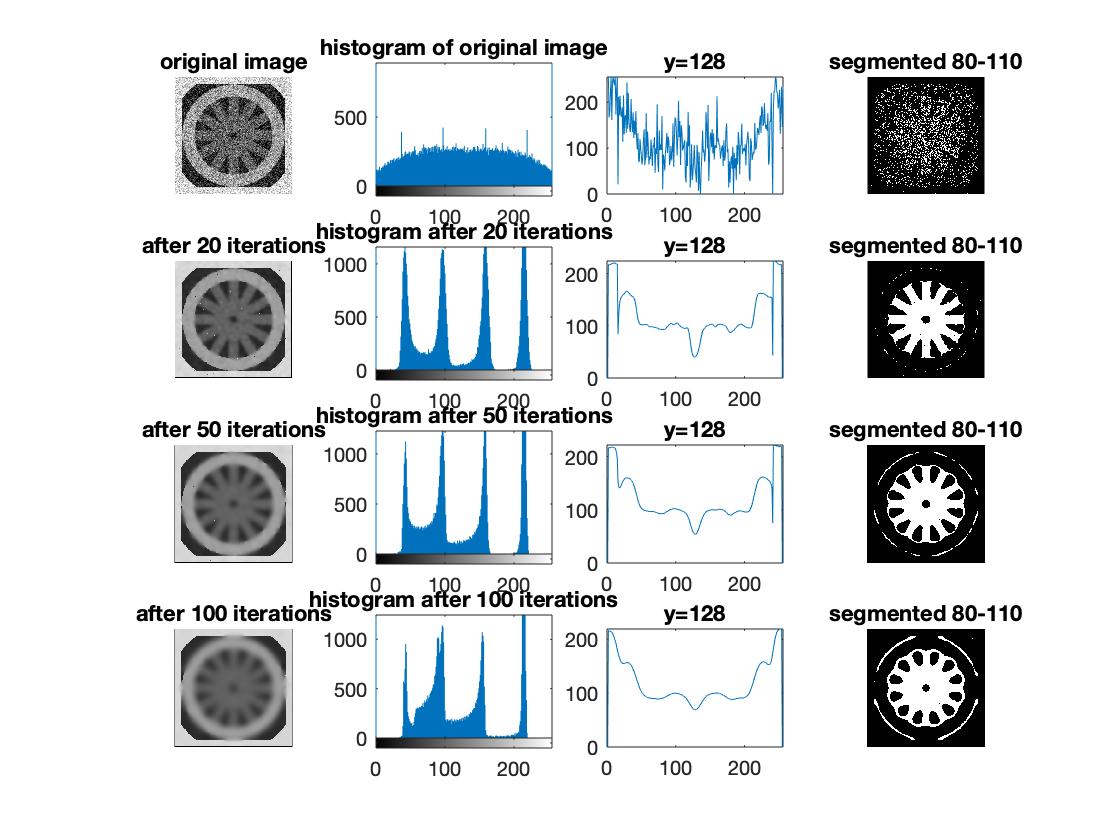
F

Figure 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.

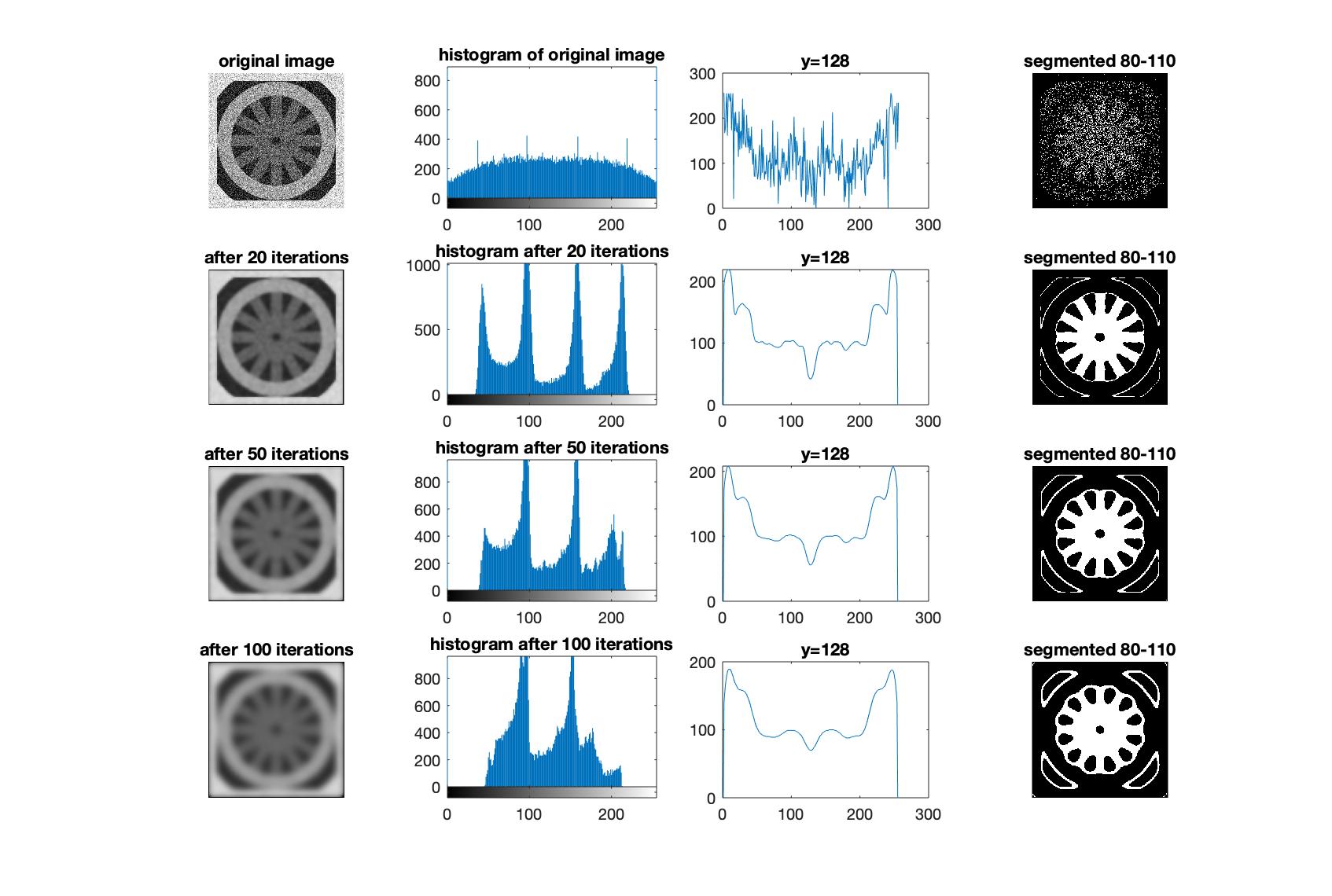
|  |  |  |
| --- | --- | --- |
|  | Mean | Std |
| 5x5 mean filter | 188.69792 | 22.589779 |
| 5x5 median filter | 191.80400 | 21.022482 |
| 5x5 alpha-trimmed mean filter | 190.12857 | 22.155251 |
| 5x5 sigma filter | 192.49910 | 21.504250 |
| 5x5 symmetric nearest-neighbor mean filter | 190.12857 | 22.155251 |

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

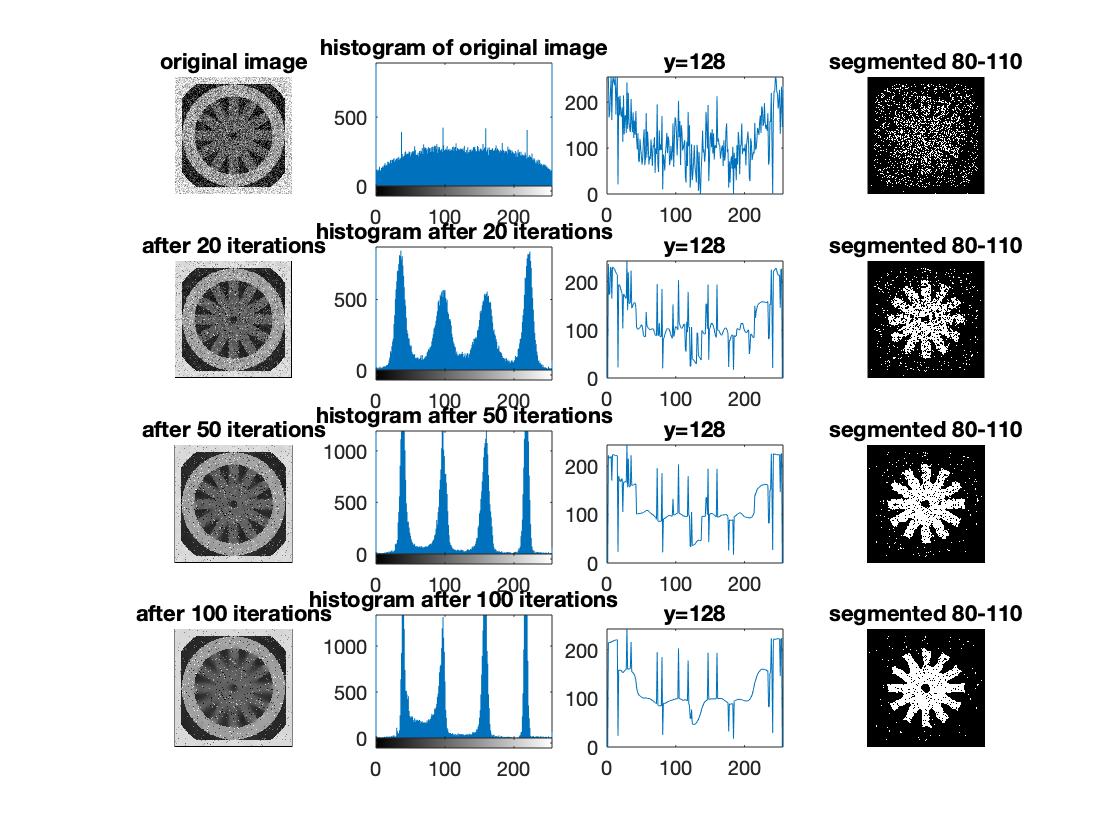
**2. Results of Anisotropic Diffusion for Image Filtering:**



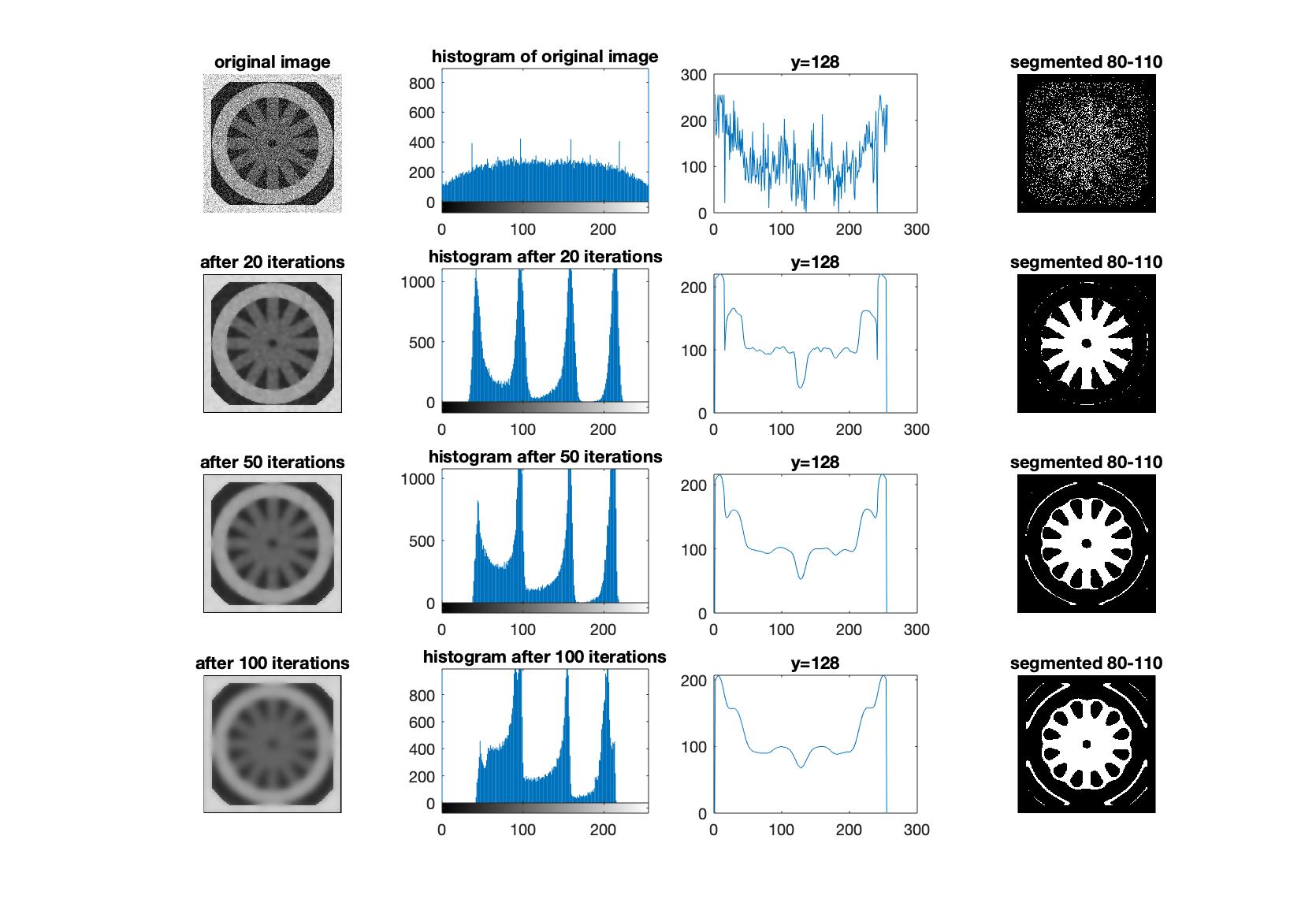
**k=50, g() = exp.**



**k=50, g() = rev**

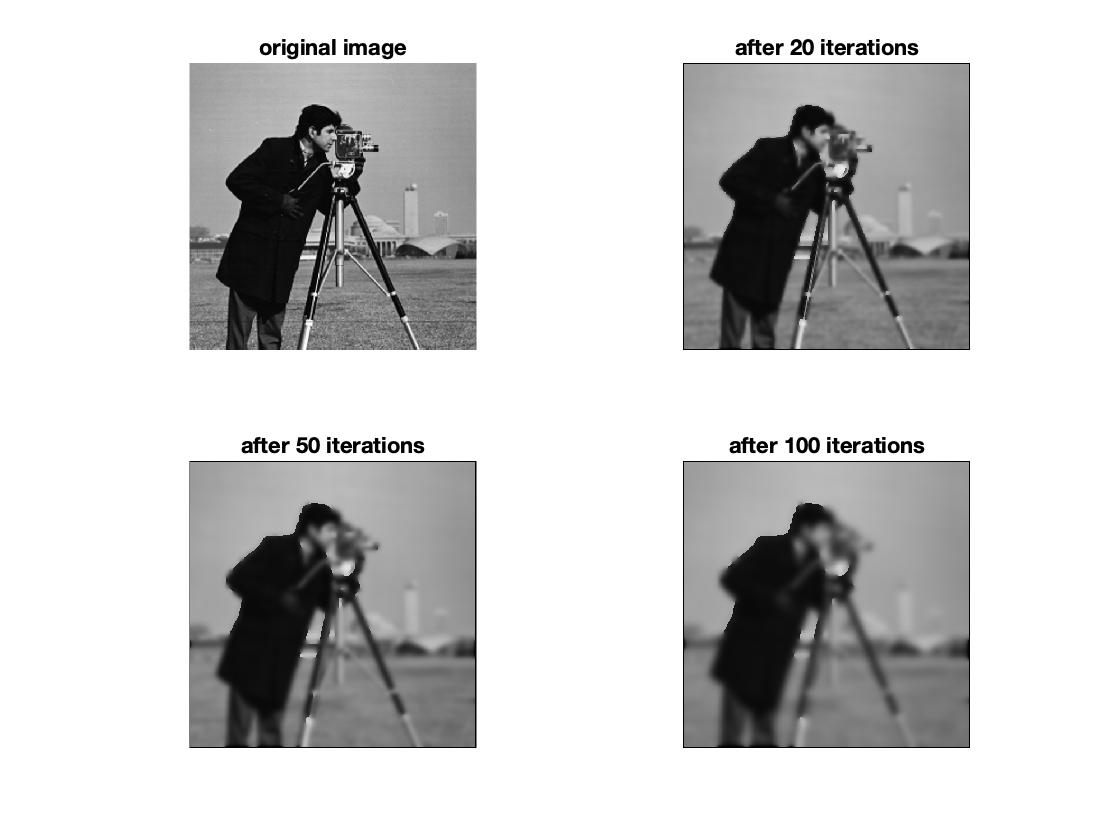


**k=25, g() = exp.**

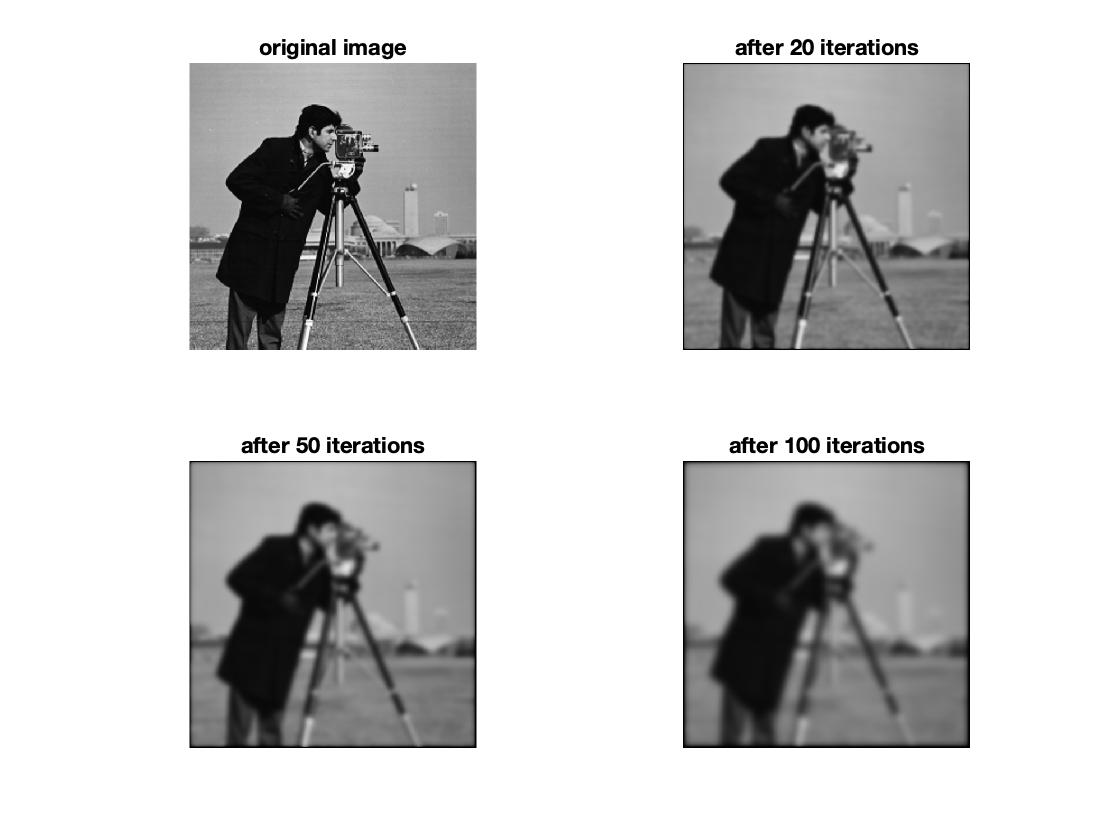


**k=25, g() = rev.**

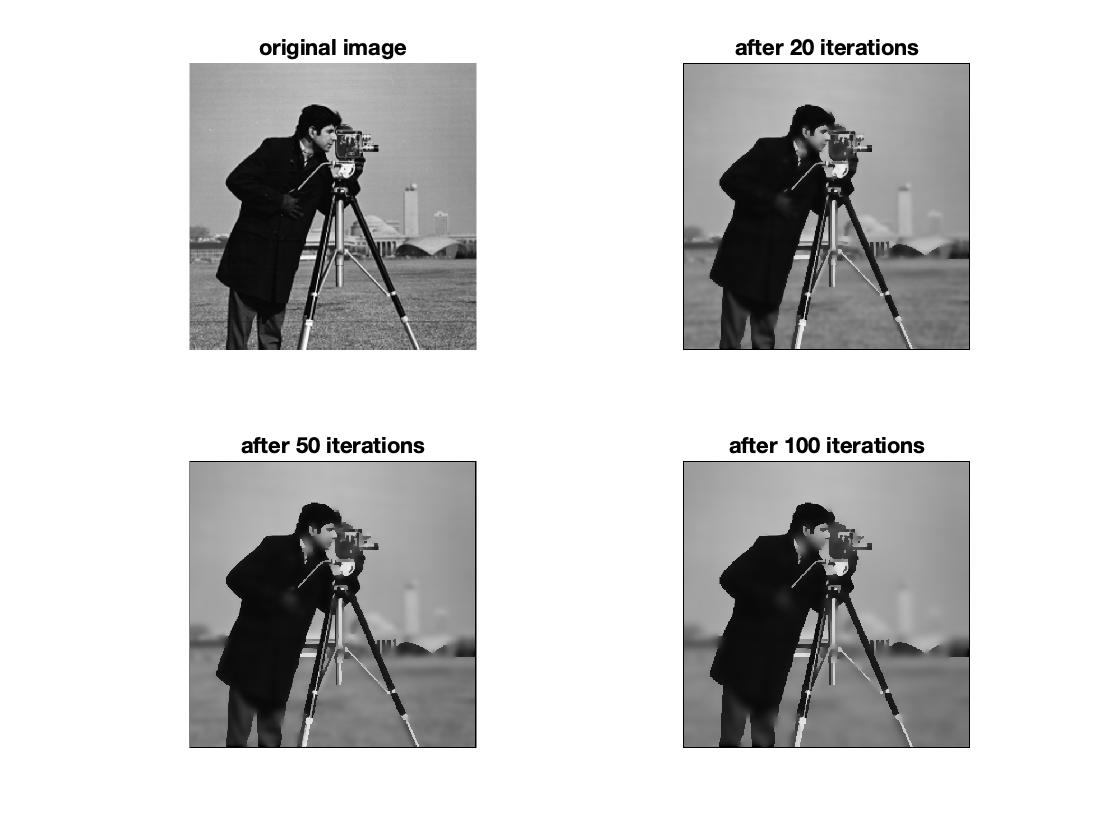
**Observations:**



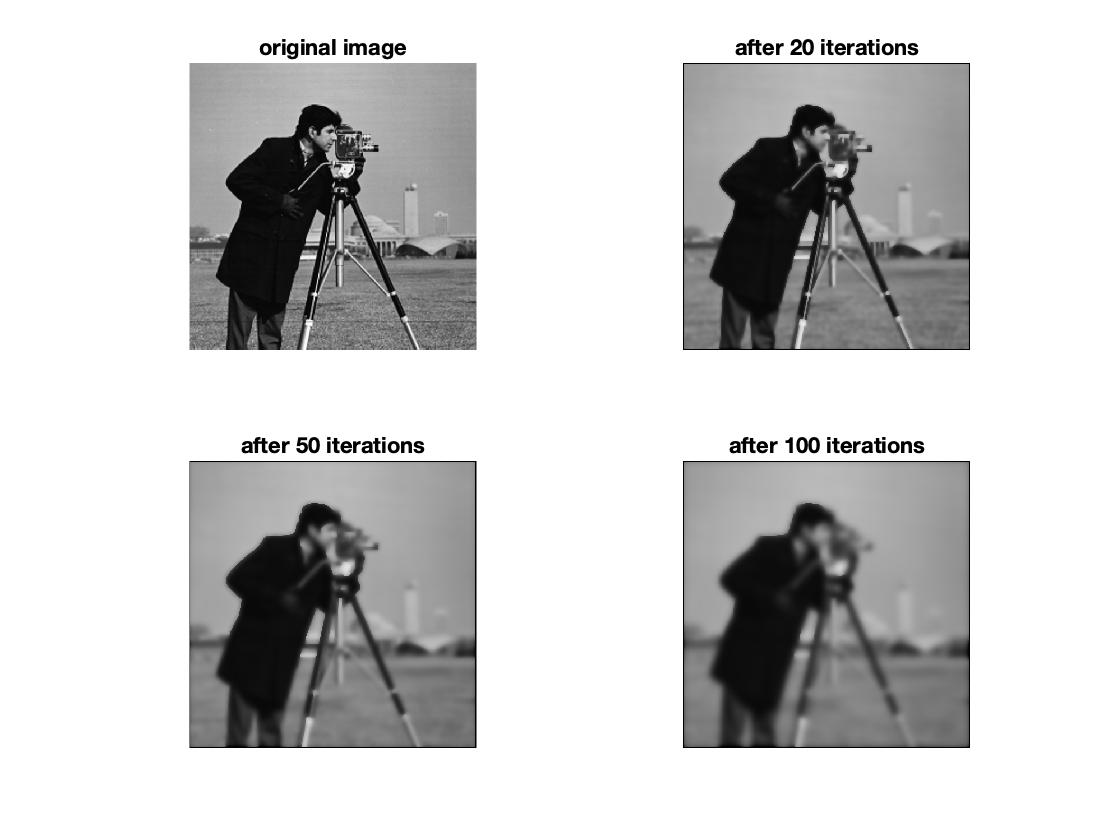
**k=50, g() = exp.**



**k=50, g() = rev.**



**k=25, g() = exp.**



**k=25, g() = rev.**