

SUSTech_EE326_lab6

Topic: Image Restoration

Author: 11911521 钟新宇

Project: lab6 report for Digital Image Processing

1. Introduction

The simple image restore process can be modeled as a linear system of degradation and recovery. This is shown below.

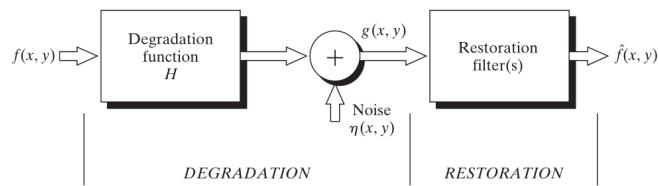


Fig 1: A model of the image degradation / restoration process.

where $f(x, y)$ represents the clean image. $f(x, y)$ is passed through a degenerate system with a frequency response of H , and superimposed with additive noise $n(x, y)$ to obtain the image $g(x, y)$ that we actually see. The purpose of image reconstruction is to remove the interference of noise $n(x, y)$ and then to obtain the estimated image $\hat{f}(x, y)$ by recovering the filter.

The relationship in spatial domain is as below.

$$g(x, y) = h(x, y) \star f(x, y) + \eta(x, y) \quad (1)$$

And the relationship in frequency domain is as below.

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (2)$$

In this lab, we will process images containing only noise and no degradation in task 1 and degraded images without noise in task 2. In task 3, we will process degenerate images with additive white noise. Since the codes are too long, I put them in the appendix file named *SUSTech_EE326_lab6_Appendix.pdf*.

2. Task 1: Remove the noise

The principal sources of noise in digital images arise during image acquisition and/or transmission. The performance of imaging sensors is affected by a variety of factors, such as environmental conditions during image acquisition, and by the quality of the sensing elements themselves. For instance, in acquiring images with a CCD camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are corrupted during transmission principally due to interference in the channel used for transmission. For example, an image transmitted using a wireless network might be corrupted as a result of lightning or other atmospheric disturbance.

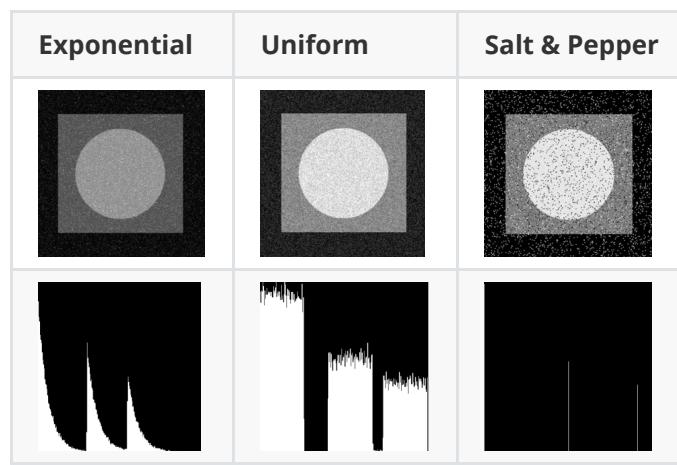
In this section, we will review the types of white noise learned and the filters used to remove the noise.

2.1 Noise Models

We use the probability density distribution (PDF) to distinguish different kinds of noise. The common noises are summarized in the following table.

| Noise | PDF | Histogram |
|---------------------------------|---|-----------|
| Gaussian Noise | $p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$ | |
| Rayleigh Noise | $p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$ | |
| Erlang (Gamma) Noise | $p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$ | |
| Exponential Noise | $p(z) = \begin{cases} ae^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$ | |
| Uniform Noise | $p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases}$ | |
| Impulse (Salt-and-Pepper) Noise | $p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$ | |

In the image, we can select a region with a uniform intensity distribution and draw a histogram of this region, then we can observe what noise is contained in the image.



2.2 Noise Reducing Filter

The researchers designed a variety of air domain filters and frequency domain filters to eliminate the noise. The common filters are shown below.

- Mean Filters
 - Arithmetic mean filter
 - The arithmetic averaging filter sums all intensity values in the neighborhood and finally averages them.

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \quad (3)$$

- Geometric mean filter
 - The geometric average filter accumulates all intensity values in the neighborhood and then performs an exponential operation. In general, the geometric mean filter has a smoothing effect comparable to the arithmetic mean filter, but it loses less image detail. Besides, if the dynamic range of the image is large, we usually do logarithmic operation, but the geometric mean filter is usually not used after the logarithmic operation.

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}} \quad (4)$$

- Harmonic mean filter
 - Harmonic mean filter works well for salt noise, but not for pepper noise. It also works well for other types of noise such as Gaussian noise.

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s, t)}} \quad (5)$$

- Contraharmonic mean filter
 - Contraharmonic mean filter is good for reducing the effect of pepper or salt noise, but it can only handle one kind of noise at a time. $q>0$ handles pepper noise and $q<0$ handles salt noise. In addition, it destroys image details: $Q>0$ causes black areas to shrink and white areas to enlarge; $Q<0$ causes white areas to shrink and black areas to enlarge.

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q} \quad (6)$$

$$f(x, y) = \frac{\text{median}_{(s,t) \in S_{xy}} \{g(s, t)\}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q} \quad (6)$$

- Order-Statistic Filters

Order-Statistic Filter is a spatial filter whose response is based on ordering (sorting) the pixel values contained in the image area enclosed by the filter. The sorting result determines the response of the filter.

- Median filter

- The median filter updates the centroid intensity value using the median of the intensity values in the neighborhood. It is particularly effective against pretzel noise and does not cause blurring of the image edges or change the image shape size. But it is not effective against uniform noise.

$$\hat{f}(x, y) = \text{median}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (7)$$

- Max filter

- The max filter updates the centroid intensity value using the maximum of the intensity values in the neighborhood. It is suitable for handling pepper noise, but it causes the black areas of the image to become smaller and the white areas to become larger

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (8)$$

- Min filter

- The min filter updates the centroid intensity value using the minimum of the intensity values in the neighborhood. It is suitable for handling salt noise, but it causes the white areas of the image to become smaller and the black areas to become larger

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (9)$$

- Midpoint filter

- Mid point filter updates the center point intensity value using the minimum and maximum values of the intensity values in the neighborhood. It combines the advantages of the max filter and the min filter to handle pretzel noise.

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right] \quad (10)$$

- Alpha-trimmed mean filter

- The alpha-trimmed mean filter removes d pixels with lower intensity values and d pixels with higher intensity values in the neighborhood, and then updates the intensity value of the centroid using the arithmetic mean of the remaining intensity values.

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} g_r(s, t) \quad (11)$$

- Adaptive Filters

- Adaptive, local noise reduction filter

- The adaptive mean filter is equivalent to a weighted average of the original image and the arithmetic mean filter result, with the weight determined by the noise variance and the neighborhood variance. Since the noise added image is equal to the original image plus noise, the neighborhood variance must be greater than or equal to the noise variance. 1. The noise variance is equal to 0, and the original image is output. 2. The neighborhood variance is equal to the noise variance, and the arithmetic mean is output. 3. The neighborhood variance is greater than the noise variance, which means that the neighborhood contains the effective information of the image, and the output image should be close to the original image (if the noise variance is small, then close to the original image; if the noise variance is large, then close to the arithmetic mean).

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_L^2} [g(x, y) - m_L] \quad (12)$$

- - Adaptive median filter
 - Adaptive median filter is suitable for pretzel noise, which ensures that the output value is not pulsed as much as possible. It has two states, as shown below.

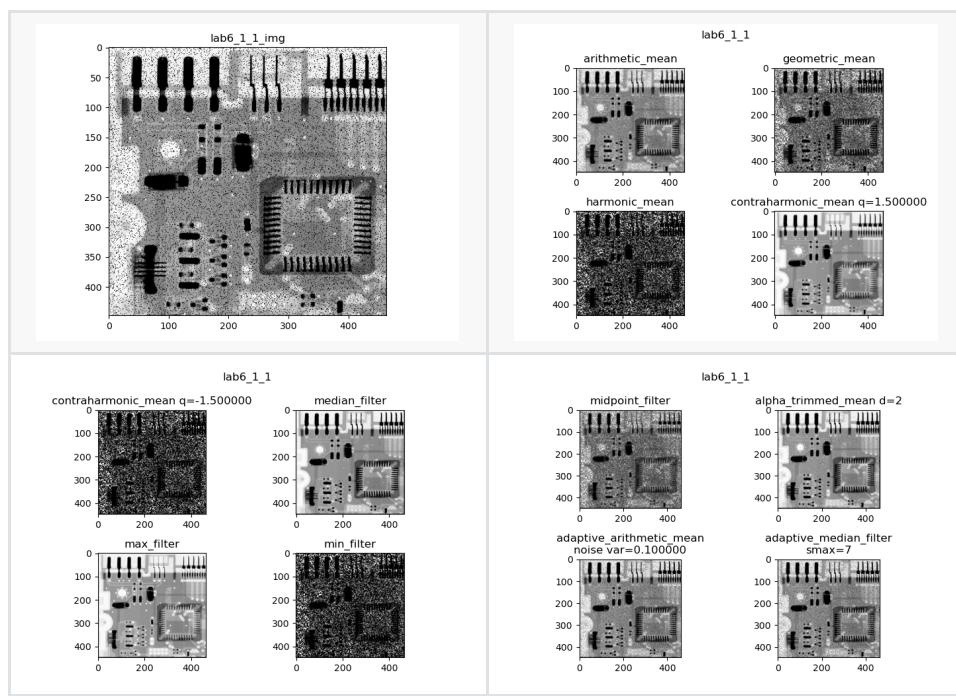
$$\begin{aligned}
 z_{\min} &= \text{minimum intensity value in } S_{xy} \\
 z_{\max} &= \text{maximum intensity value in } S_{xy} \\
 z_{\text{med}} &= \text{median of intensity values in } S_{xy} \\
 z_{xy} &= \text{intensity value at coordinates } (x, y) \\
 S_{\max} &= \text{maximum allowed size of } S_{xy} \\
 A1 &= z_{\text{med}} - z_{\min} \\
 A2 &= z_{\text{med}} - z_{\max} \\
 B1 &= z_{xy} - z_{\min} \\
 B2 &= z_{xy} - z_{\max}
 \end{aligned} \quad (13)$$

- We use two conditions to determine the states.
 1. $a1 > 0$ and $a2 < 0$: This condition determines if the median is a pulse by comparing the relationship between the median and the maximum and minimum values in the neighborhood. If the condition is satisfied, it is not an impulse and goes to State B. If it is an impulse, then the window size is increased. If the window is increased to the maximum and the median is still a pulse, then the median is output directly.
 2. $b1 > 0$ and $b2 < 0$: This condition determines whether the point being processed is a pulse by comparing the original pixel with the maximum and minimum values in the neighborhood. If the condition is satisfied, it means it is not a pulse, and the gray value of the original pixel point is output. If it is a pulse, the median value of the neighborhood (not a pulse) is output, which is equivalent to median filtering.

2.3 Results

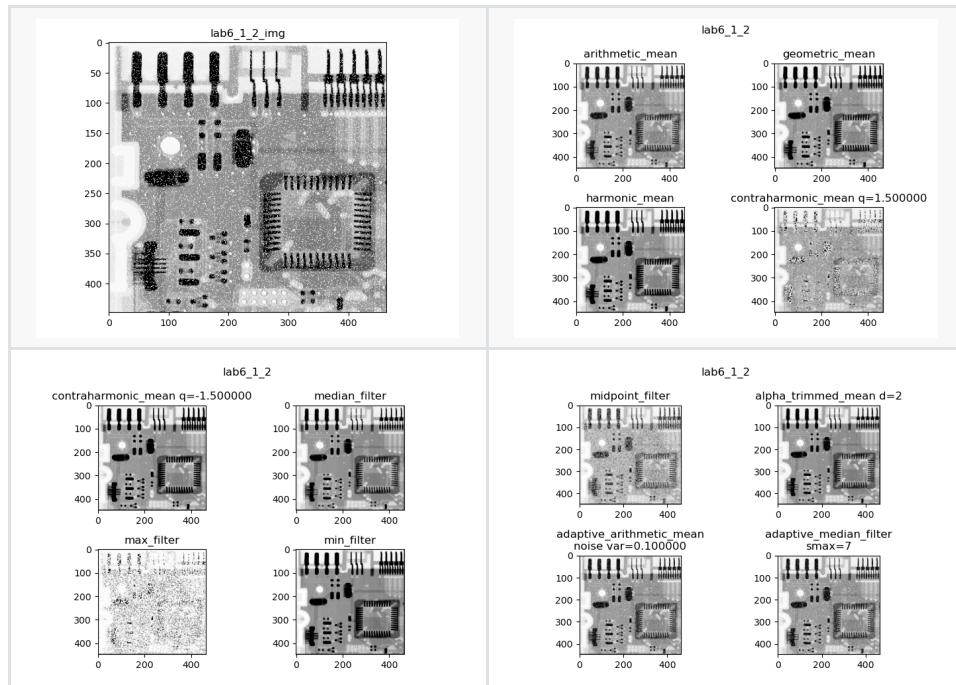
2.3.1 Image 1

We found that this image contains only pepper noise with a noise variance of 0.1, so we presume that the Contraharmonic mean filter with Q greater than 0, median filter, and max filter have better filtering effects. The result is also the same.



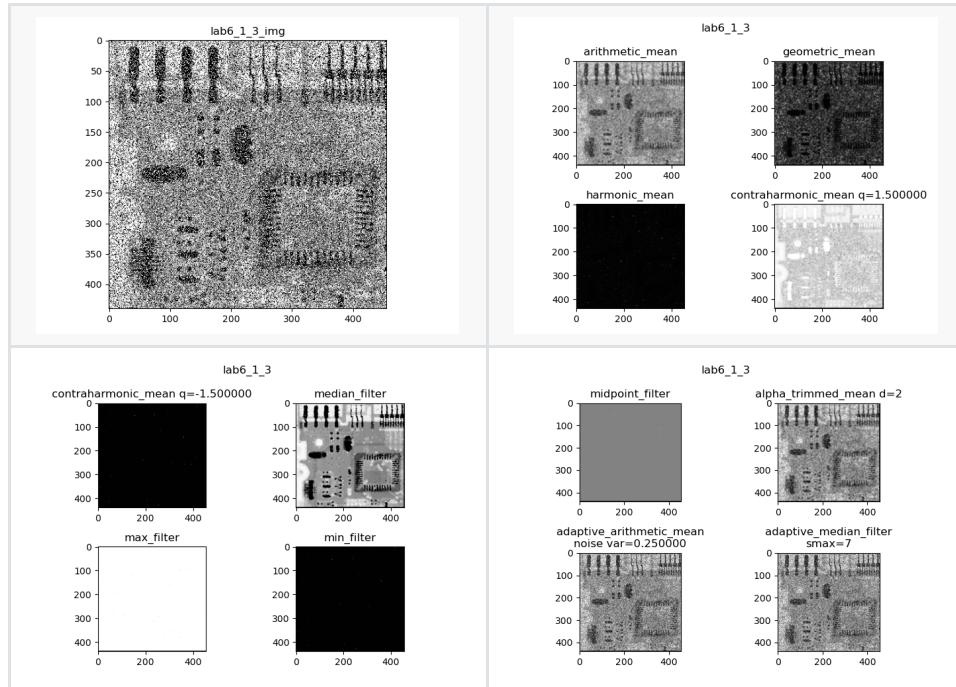
2.3.2 Image 2

We find that this image contains only salt noise with a noise variance of 0.1, so we presume that the contraharmonic mean filter with Q less than 0, median filter, and min filter have better filtering effects. The result is also the same.



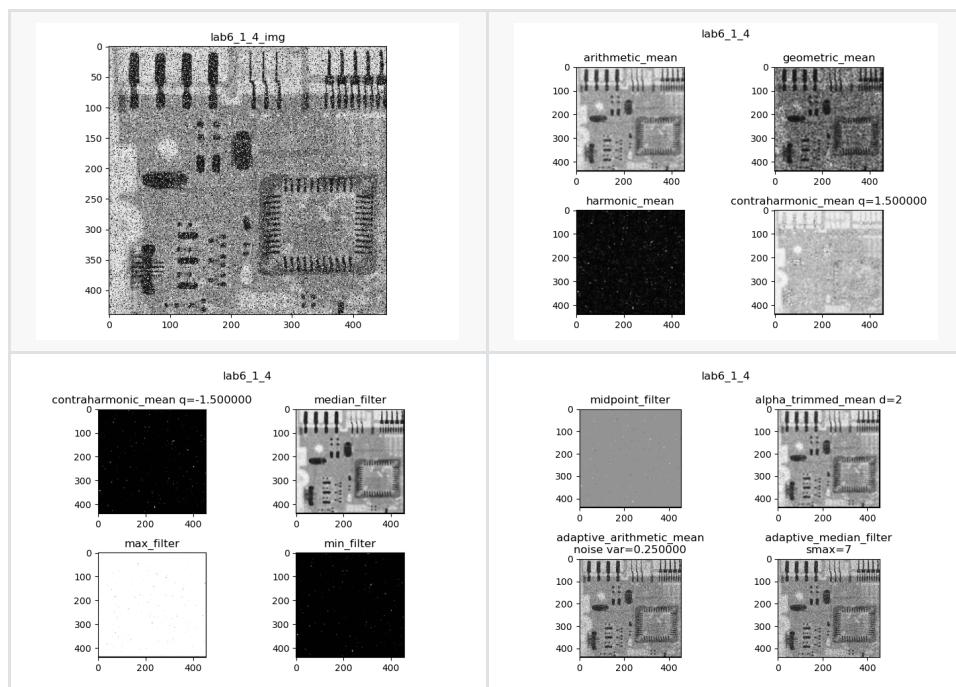
2.3.3 Image 3

We find that this image contains both pretzel noise and noise variance of 0.25. Therefore, we speculate that the max / min filter, mid point filter, harmonic / contraharmonic mean filter which can only handle the same kind of noise, will not work. The results are the same, only the median filter works best, and the adaptive filter works better.



2.3.4 Image 4

We find that this image contains both pretzel noise and uniform noise. The median filter, adaptive filter, and arithmetic average filter all work well in this case.



3. Task 2: Restore the image

3.1 Degradation function

According to the previously introduced image degradation/restore model, the relationship between the restored and degraded images in the frequency domain is shown below.

$$H_s(u, v) = \frac{G_s(u, v)}{\hat{F}_s(u, v)} \quad (14)$$

In order to recover the image, we need to design a restore filter, which is the inverse filter of the degradation filter, and its frequency response is the inverse of the frequency response of the degradation filter. Therefore, in order to design the restore filter, we first need to obtain the frequency response H of the degenerate filter. Fortunately, researchers have summarized the degenerate functions of atmospheric turbulence and uniform motion of the camera.

- Atmospheric turbulence

$$H(u, v) = e^{-k(u^2+v^2)^{5/6}} \quad (15)$$

- Motion blurring

$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin^{\pi(ua + vb)} e^{-j\pi(ua + vb)} \quad (16)$$

3.2 Restore Filters

In this lab, we learn about full inverse filter, radius-limited inverse filter, and Wiener filter. They are all frequency domain filters.

- full inverse filter

The full inverse filter is very simple, it assumes that the degraded image is not disturbed by noise, then the relationship between the frequency response of the restored image, the degraded image, and the degraded filter is shown below.

$$\hat{F}_s(u, v) = \frac{G_s(u, v)}{H_s(u, v)} \quad (17)$$

Therefore, the full inverse filter serves to transform these three matrices into the frequency domain, perform the division operation, and then transform back into the spatial domain.

- radius-limited inverse filter

Based on the fully inverse filter, the radius-limited inverse filter puts a limit on the size of the frequency, which is equivalent to a series of a fully inverse filter and a low-pass filter.

- Wiener filter

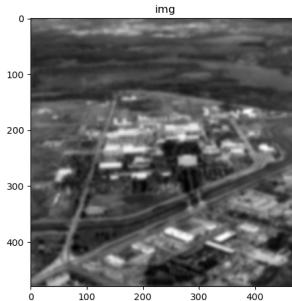
The Wiener filter is also known as the minimum mean square error filter. It minimizes the mean square error of the original and recovered images. The minimum of the error function is given in the frequency domain by the expression.

$$\hat{F}(u, v) = \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v) \quad (18)$$

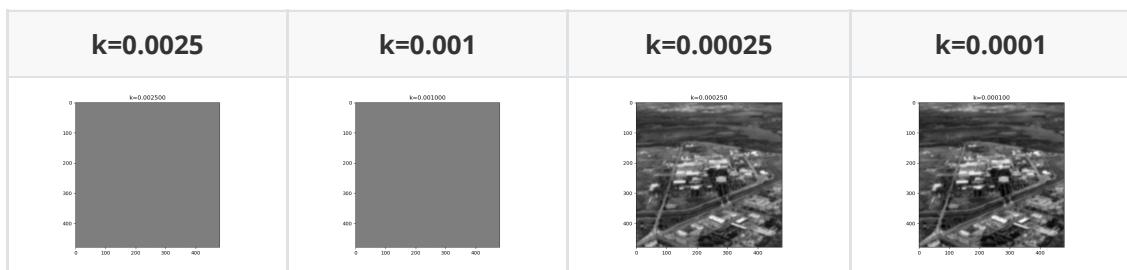
where $K = S_\eta(u, v)/S_f(u, v)$ is a constant.

3.3 Results

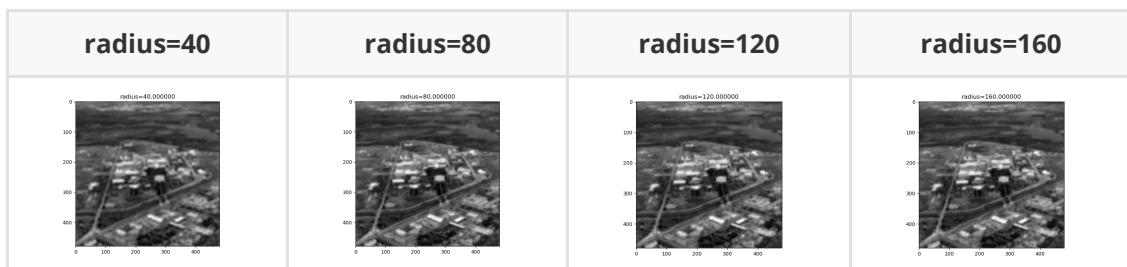
- The original image



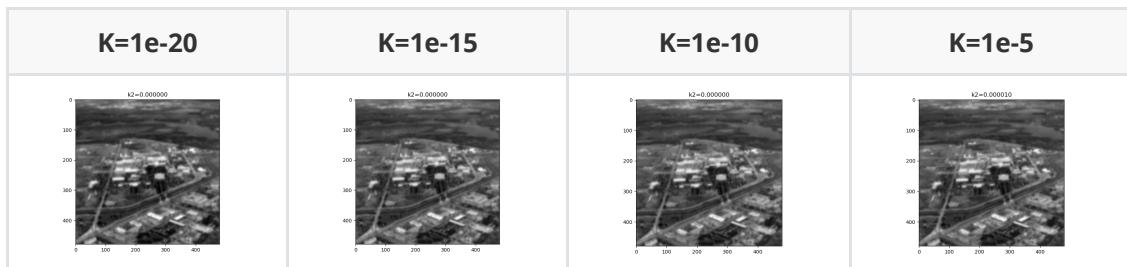
- full inverse filtering



- radius-limited inverse filtering ($k=0.00025$)



- Wiener filtering ($k=0.00025$)



We can observe that the best result is obtained with the Wiener filter for $k=0.00025$ and $K=1e-15$.

4. Summary

In this lab, I learn about common noise in digital image processing, and the various filters used to remove noises. I also learn about image restoration methods, and learn about image degradation functions and restore filters.

