Lab8：Image Filters, Part I

Mingqin Dai

I. Motivation

Digitized images are often corrupted by noise from several sources. We are doing this lab to present some insights into the problem and to describe some simple and common techniques for enhancing images corrupted by noise, linear and nonlinear filters. These are used to solve image processing problem that helps discriminate between image and noise, which is image denoising. Noise degradation is bothersome in applications. Image denoising is the manipulation of the image data to produce a visually high quality image, which improve images by removing as much noise as possible without degrading the image itself.

II. Method

Making a lot of noise

In this lab, we first corrupt the image *lena* using two types of noise, which are additive Gaussian noise and salt and pepper noise. The images that corrupted with these noise are as follows.

Lena_s (salt and pepper noise)

Lena_g (additive gaussian noise)

Salt and pepper noise is a set of impulses randomly placed on the image plane. We can see it in the image that this noise contains random sparsely occurrences of black and white pixels. Additive Gaussian noise has a normal distribution in the time domain with an average time domain value of zero. We can see it in the image that it causes variations in intensity of pixels in the original image drawn from a Gaussian normal distribution. ***(describe in your report how each type of noise affects the image quality)***

Part A. Linear Filtering

We apply Gaussian filter with different window size and main lobe size and Laplacian of Gaussian filter with different window size and standard deviation to remove the Additive Gaussian noise and Salt and peper noise respectively, the result are as follows.

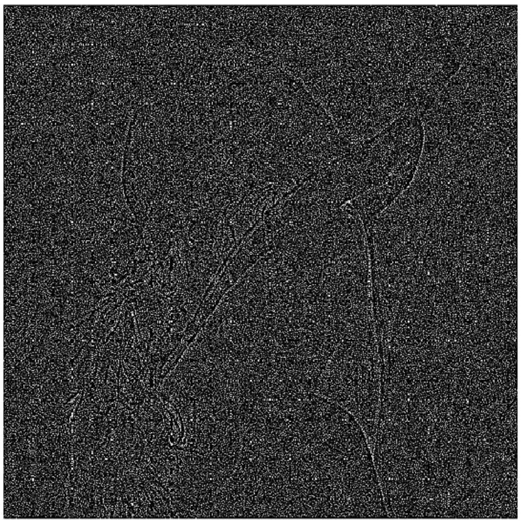
a. Denoising Additive Gaussian Noise with Gaussian filter

	Window size: 5x5	Window size: 10x10
Lobe size: 0.25		
Lobe size: 0.5		
Lobe size: 1.0		

b. Denoising Salt & pepper noise with Gaussian filter

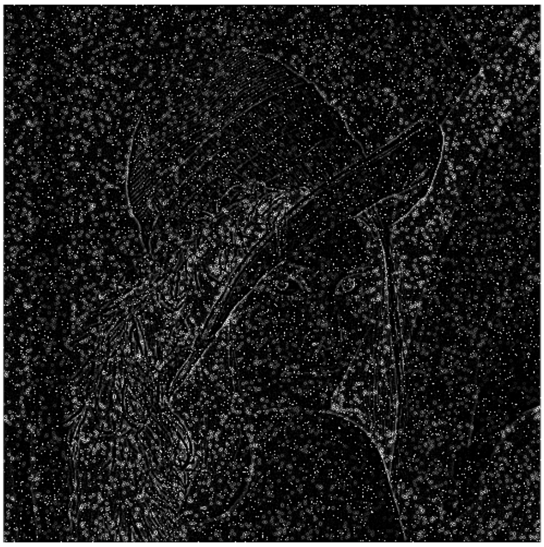
	Window size: 5x5	Window size: 10x10
Lobe size: 0.25		
Lobe size: 0.5		
Lobe size: 1.0		

c. Denoising Gaussian noise with Laplacian of Gaussian filter

	Window size: 5x5	Window size: 10x10
Standard deviation: 0.5		
Standard deviation: 1.0		

The image on the left is got by applying Laplacian of Gaussian filter directly to image *lena* (without noise). The Laplacian of Gaussian filter can be thought of as the "second derivative" of the Gaussian filter. The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection (see zero crossing edge detectors). Thus the Laplacian of Gaussian filter is useful in edge detection.

d. Denoising salt & pepper noise with Laplacian of Gaussian filter

	Window size: 5x5	Window size: 10x10
Standard deviation: 0.5		
Standard deviation: 1.0		

The above images are results we got. Overall, these two filters are more effective at removing Gaussian noise, though it could not remove all the noise and it actually blurs the images. For Gaussian filter, as window size or lobe size increase, more details and noise has been removed and the image become more blur. For log filter, as the window size increase, more noise has been removed. With standard deviation equals to one, we could hardly recognize the edges, however, with window size 5x5, filter with deviation equals to one actually removes more noise apparently. These two filters could not remove salt and pepper noise and it blurs the image with the increasing of parameters. We can still recognize salt and pepper noise in all output images. (Expect image filtered by Laplacian of Gaussian filter with window size 10x10 and deviation 1.) The salt and pepper noise has been blurred as lobe size and window size of Gaussian filter increases. (Though we can still recognize the noise) ***(summarize your observations in your report.)***

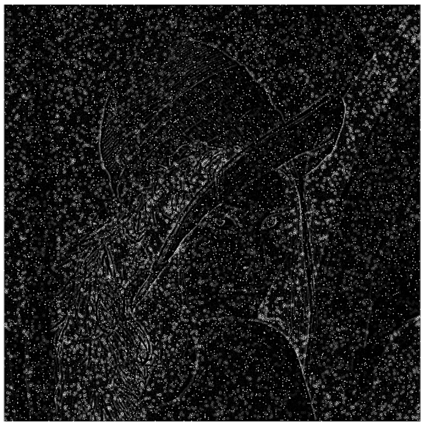
After applying a Gaussian filter to the noisy images, we can find from the above e results, the larger the mask, the more noise reduction we can obtained, but the more edges been blurred. After applying a log filter to the noisy images for the Gaussian noise, based on the above results, we can hardly recognize the noise for window size of 10x10 and deviation 0.5, the noise has been effectively removed. But the larger the deviation, the edges are more blurred which could hardly be recognized. Comparing these two filters, if we aimed at detecting edge, the log filter is preferred with window size 10x10 and standard deviation 0.5. (This is the proffered result in the four result we get.) Otherwise, the Gaussian filter is preferred, subjectively, the filter with 5x5 window and lobe size 1.0 giving the best result among the six filters. It reduces noise in the condition that not blur the edges such apparently. ***(Q1)***

In time domain, linear filter replaces the original pixel by the weighted sum of its neighboring pixels and the weights are the filter coefficients. The output image is convolution of original image and linear filter. And median filter replaces the original pixel be the median pixel value in the window, it can’t be represented by a convolution. In frequency domain, convolution in spatial domain is equivalent to component-wise multiplication in frequency domain, while a median filter because the transforms that we use for converting between spatial and frequency domains are only valid for linear systems. Thus median filter has no frequency domain equivalent, we can't perform median filtering in the frequency domain. Based on the above properties in time and frequency domain, linear filter generally expects pixels to “be like” their neighbors, where surfaces turn slowly and relatively few reflectance changes. In this case noise corrupts all pixels is effectively suppressed, and as the parameter goes up in Gaussian filter, the image gets smoother and more blurred. Thus linear filters effective for removing noise in some cases where median filters are not and vice versa. A linear filter is effective in cases where the input data contains a large amount of noise but the magnitude is low. It is much effective when the noise corrupts all pixels and very large noise values are unlikely exists, i.e. Gaussian noise. Conversely, if an image contains a low amount of noise but with relatively high magnitude, i.e. salt & pepper noise, then a median filter may be more appropriate. Median filtering is remarkably effective when the noise corrupts isolated pixels. ***(Q2)***

Part B. Median Filtering

We apply median filter to image with salt and pepper noise and image with Gaussian noise. In this lab, a median filter computes the median value in an 3x3 block and sets the center pixel of the block to the median value *a.Comparasion between lena_s_median, lena_s_smoothed, lena_s_logged.*





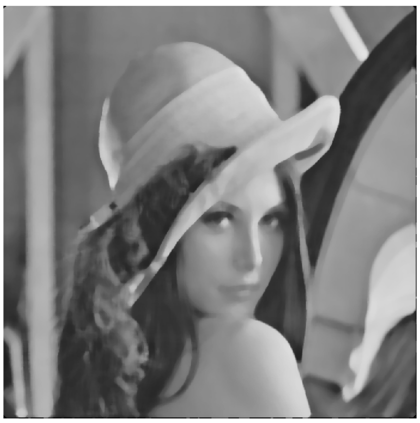
lena_s_median　　lena_s_smoothed　　lena_s_logged

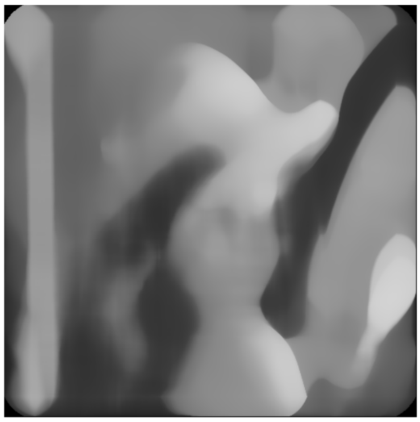
Compared with Gaussian filter and Laplacian of Gaussian filter, Median filter remove the salt and pepper noise effectively.

b.Vary the median filters size



filter size: 2x2　　3x3　　5x5





filter size: 11x11　　21x21　　51x51

When the filter size is smaller than 3, there exits unremoved salt and pepper noise. And as the filter size increases, the image become more blur, however, the noise has been removed. ***(describe in***

your report how it affects the image.)

c.Applying median filter to lena_g

lena_g filter size: 2x2 3x3

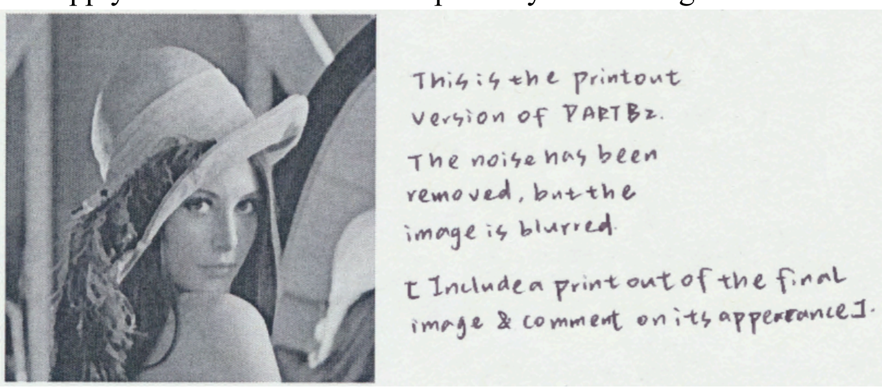
  

filter size: 5x5 11x11 21x21

After applying median filter with different size to image with Gaussian noise, we can recognized that the noise does not removed. It does not remove the Gaussian noise as effective as Gaussian filter does. And as the filter size increases, the image become more blur, the edges become less apparent, which means the median filter tens to round the "corners" in the image and does not preserve the original image information.

Part B2: Ad infinitum

We apply the 3x3 median filter repeatedly to the image 50 times. The results are as follows.



This is the version displayed on computer, with 50, 100, 500, 1000 times loop.



Loop times: 50 times 100 times 500 times

Loop times: 1000 times

After applying the median filter to the image with salt and pepper noise 1000 times, it gives the same image as applying the median filter to the image with salt and pepper noise 500 time. Under repeated application of a median filter to an image, the image converges to an invariant image. For 1-D case, for any finite length, M level, signal that ends with constant regions of length N+1 will converge to an invariant signal in a finite number of passes of this median filter. Such an invariant signal is called a root. In 2-D case, the image will also converge to an invariant image, which is the root image of a 2-D median filter. ***(Q3)***

III.Conclusion

For the results of this lab, linear filter is much effective in removing noise that corrupts all pixels and very large noise values are unlikely exists, i.e. Gaussian noise. Conversely, median filter is remarkably effective in removing noise corrupts isolated pixels, i.e. Salt and pepper noise. Median filter with larger size will result in more blur image. And under repeated application of an identical median filter to an image, the image converges to an invariant image. I learned that linear and nonlinear filters are the two most utilized forms of filter construction for image denoising. Selecting of different types of filter is depending on the image data. If an image contains a low amount of noise but with relatively high magnitude, a median filter may be more appropriate. If the input data contains a large amount of noise but the magnitude is low, a linear filter is more appropriate. And a median filter is a nonlinear digital filter which consists of a window that moves over an image of finite length, for each input sample, the corresponding output point is the median of all samples in the window centered on that input sample. After applying a median filter several times, the image will converge to an invariant image, which is the root image of a 2-D median filter.