Digital Image Processing HW #4

0310139 曾敏原

Input 1 - Gaussian Blur

Result:

Blurred Image	Deblurred Image	Original Image
		
PSNR: 65.29286159552628	PSNR: 67.77563641690412	-

Analysis:

In this problem, the input is a gaussian blurred image with unknown sigma. Before performing deconvolution of blurred image, we need to know the **sigma of the gaussian kernel** at first.

The following statement is to explain how I find the sigma of the gaussian kernel. I use the method mentioned in [1] and [2]. The brief concept of them is to collect several differences of blurred image and convolution of blurred image and a special sigma gaussian kernel. Fig.1 is the pseudo-code for generating sigma set and equation (1) is to calculate the difference above.

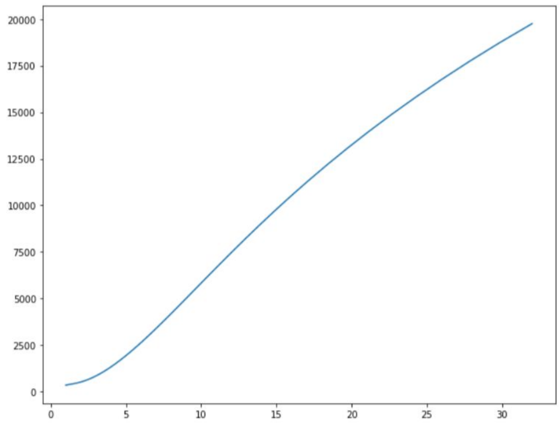
```
octaveDivisions = 10
numOfOctaves = 5
scaleFactor = 2.0^(1.0/octaveDivisions)
numOfLevels = octaveDivision*numOfOctaves+1
sigma(1) = 1

For s = 2 to numOfLevels
      Sigma(s) = sigma(s-1)*scaleFactor
end
```

Fig 1. Pseudo code

$$E(\sigma_2) = \sum \left| F \otimes G\left(\sigma_1^2\right) - F \otimes G\left(\sigma_1^2\right) \otimes G\left(\sigma_2^2\right) \right|,$$

Equation (1)

Now, define equation (1) as error. Therefore, we can collect an error list and its corresponding sigma set. The following plot shows the curve of error list and sigma set.



(where x axis means sigma and y axis means error)

After generating this curve, I find the predicted sigma value with get local extrema by calculate finite differences which is mentioned in [1] and [2]. After I perform this procedure on input 1, I get the sigma equal to 9.18958683997627.

Algorithm:
In order to restore the image, I use the typical degradation model in Fig. 2 and Wiener Filter(equation 2).

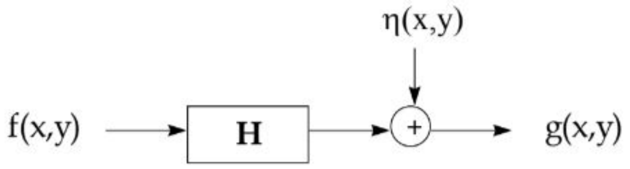
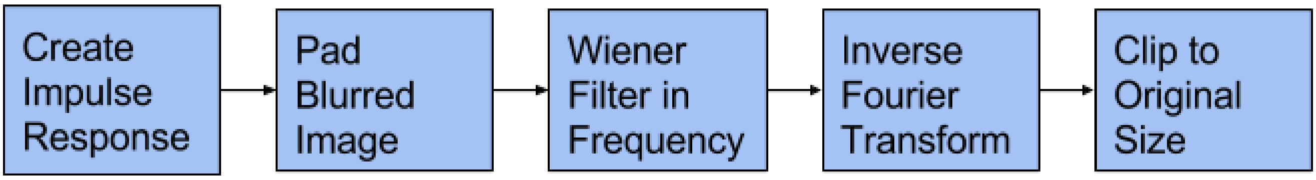


Fig 2. degradation model

$$\hat{F}(u,v) = [\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2}]G(u,v)$$

Equation 2. where P is fourier transform of a laplace filter

Here is procedure of my algorithm:



Create Impulse Response

To restore the image, we need to know **H** in Fig 2. first. With a known sigma, we can create a gaussian kernel(which is known as **h**) easily**.** Then convert **h** to frequency domain to bulid **H**. For Input 1, I create a gaussian kernel(Fig 3.) with size(71, 71) and sigma equal to 9.18958683997627.

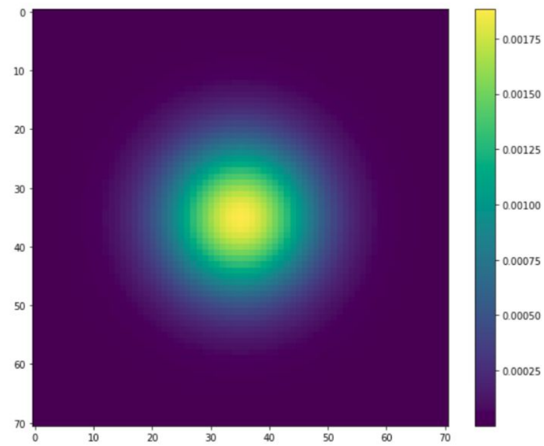


Fig 3. Gaussian Kernel

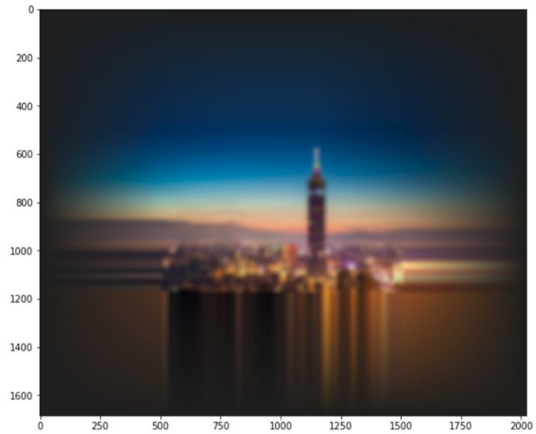


Fig 4. Padded Image

Pad Blurred Image

In order to reduce distortion between edge of image after DFT, I use "Linear Ramp" padding algorithm to pad the blurred image. The result is like Fig. 4.

Wiener Filter in Frequency Domain

I perform DFT on gaussian kernel and padded image. Then use Wiener filter in Equation 2 to restore the image. In this equation, I modify the value of gamma to get the highest PSNR. When gamma is equal to 1, I get the best PSNR.

Inverse Fourier Transform and **Clip to Original Size**

After restoration in frequency domain, I perform inverse DFT to show image. However, the image is padded so I clip the image to original size.

Discussion:

Why Should we Pad Image by this Way

I pad image by 'linear ramp' because I want to reduce the effort of edge effect caused by DFT. The 'zero-padding' method is not ideal in this condition because there are edges between original image and zero pad. These edges will cause serious 'ripples' after restoration. The following table is to show how padding method influence the result.

Without Padding	Zero Padding	Linear Ramp Padding

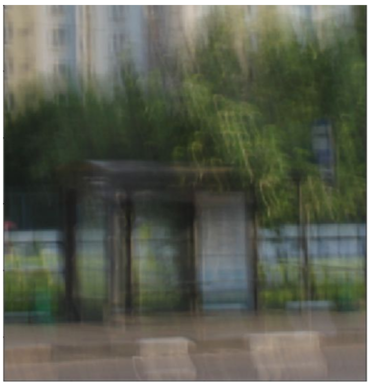
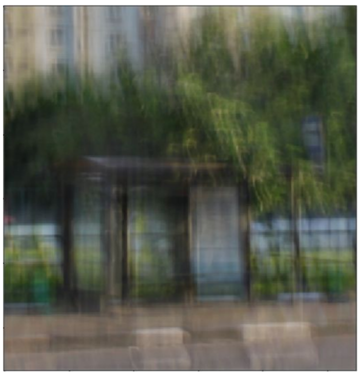
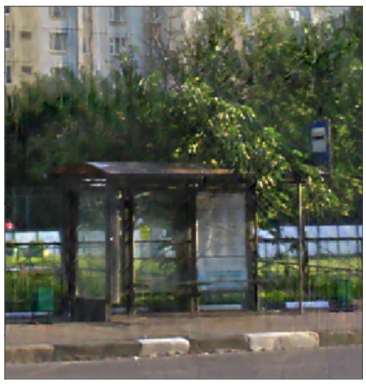
How K of Wiener Filter Influence the Result

In theory of Wiener filter, k will dominate the frequency gain when gain of inverse filter is too low. Here is some test and result on k with 'linear ramp' padding.

K=0.01	K=1	K=100
		

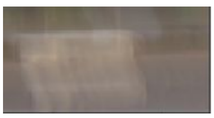
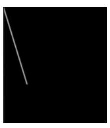
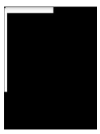
Input 2 - Motion Blur

Result:

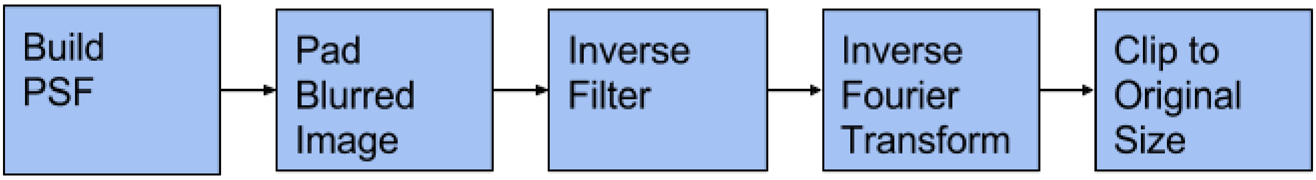
Blurred Image	Deblurred Image	Original Image
		
PSNR: 58.70392725893129	PSNR: 60.149343579100844	-

Analysis:

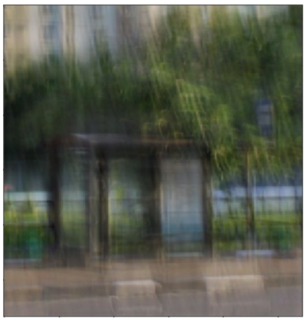
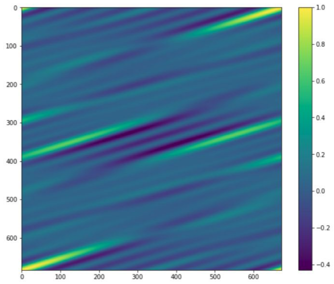
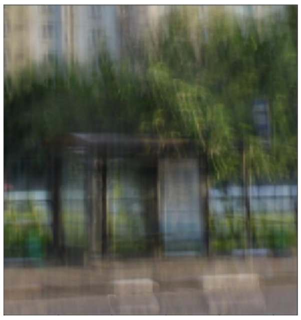
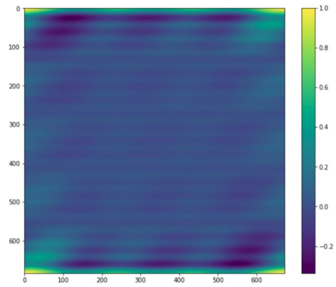
I try to estimate psf of this blurred image in order to restore it. First of all, I assume this blurred image is caused by an uniform linear motion so I can estimate kernel of filter from the blurred image. Then I try to estimate the motion from lower right part of image(Fig. 3). I get a slope which is 21/6 and draw in Fig 4. However, if I use the kernel in Fig 4. directly, I get a worse PSNR score than blurred image. Therefore, I modify the kernel to Fig 5. and get a better PSNR score.

Fig. 3	Fig. 4	Fig. 5
		

Algorithm:



This algorithm procedure is almost the same as input 1. The only difference is using inverse filter instead of wiener filter. The following table is comparision of two kernel in Fig. 4 and Fig. 5.

Kernel in Fig. 4	Spectrum of Fig. 4	Kernel in Fig. 5	Spectrum of Fig. 5
			

Discussion:

Why Predicted Filter Not Perform Well

1. In kernel prediction, I make an assumption that the blurred image is caused by uniform linear motion. From above table, I find "uniform" may be wrong in this picture because the object(lower right part of image) we assume will restore after restoration indeed become better. However, the other objects are still blurred. This phenomenon implies this image is not caused by uniform motion.
2. Distortion occurs when restoration. My restoration of the image is to move back the blurred pixel to original pixel, but this operation will also move correct pixels up to wrong pixels. This phenomenon is occured in table above.

Possible Solution:

Using blind deconvolution proposed in [3]: constructing loss function and perform optimization to get psf and deblurred image. With blind deconvolution, it could decrease the estimation error caused by human.

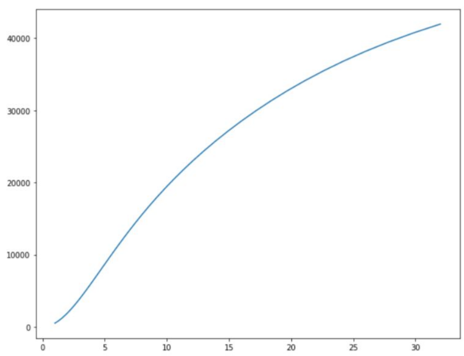
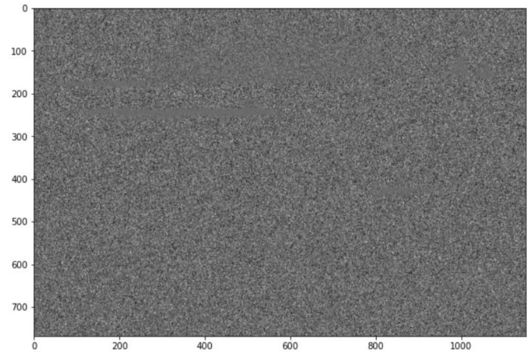
Input 3 - Gaussian Blur and Noise

Blurred Image	Deblurred Image	Original Image
		
PSNR: 58.827829808452535	PSNR: 68.20514181918242	-

Analysis:

In this part, the input is a image with gaussian blur and noise. However, if I follow the method in [1] and [2] directly, I will get a sigma is equal to 1 because of the noise. Therefore, I use a gaussian filter with sigma=3 to blur image first. Then performing method in [1] and [2] on the image. I get the sigma equal to 4.59, then get the original sigma by $sigma\_ori = \sqrt{sigma\_est^2 - sigma\_gau^2}$. The final sigma is equal to 3.48.

In [1] and [2], they also mention a method that estimate noise power sigma and noise spectrum density which can used to be K in wiener filter. However, I follow the procedures in [1] and [2] and get noise sigma is equal to 14.80. The value of noise sigma is too big to follow the procedure of [1] and [2]. Their method has best performance with noise sigma under 5.

Error plane after modification	Edge detection for noise estimation
	

Algorithm:

The procedures are the same as input 1, but the wiener filter is different from input 1's. I use constant K instead of laplace noise wiener filter that modify K and get good result. The padding trick is also used in this part.

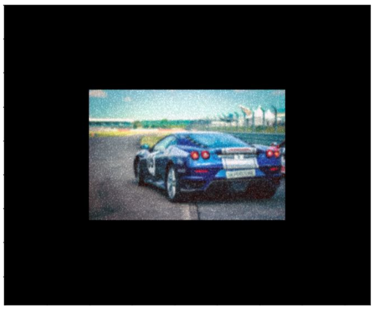
Discussion:

I will try different padding algorithms and different Ks for wiener filter like input 1 again.

A Trick for Padding an Image

Because the input image exists some noise, if I directly pad the image with "linear ramp" algorithm, it would take consideration on noise. In order to prevent it, I pad the image after passing gaussian filter with sigma=3. Then replacing the original image into center.

Different Padding Algorithm

No padding	Zero padding	Linear Ramp Padding
		
		

I find something interesting on the above table. The quality of non-padded or zero-padded image generated is as good as the quality of linear ramp padded image. I think it's because the sigma of the kernel is smaller than input 1's. The ripple influence is weaker.

Different K of Wiener Filter

K=0.0001	K=0.06	K=1
		

When K=0.0001, every small H(u, v) will be dominated by K and gain is about 10000. The noise will be amplified. When K=0.06, I get a acceptable result. When K=1, for every H(u, v) will dominated by K and gain will be 0.5 to 1.0.

Reference:

[1] Robinson,Roodt, “Blind Deconvolution Of Gaussian blurred images containing additive white Gaussian Noise ”,IEEE International Conference on Industrial Technology(ICIT), 2013, pp. 1092-1097.

[2] P.E. Robinson, Y. Roodt, A. Nel, "Gaussian blur identification using scale-space theory", Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa (PRASA), In Press, 2012.

[3] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. SIGGRAPH, 2008.