EE610 Assignment-2: Deblurring With Known Kernel

Karan Taneja*

***Abstract*— The report summarizes results of task one from assignment 2. Four kernels are used to blur the image and a noise is added. To deblur the image, the known kernels are used. Full inverse, truncated inverse, approximate Wiener filter, and constrained LS filtering is used to deblur the images. PSNR and SSIM metrics are used for evaluation.**

I. INTRODUCTION

A simple degradation model is used with a small 20x20 kernel with additive white Gaussian noise. In this task, we assume that the blur kernel is known and explore various techniques to deblur the image. Though it is rarely the case that blur kernel is known, these techniques are important nevertheless. This is because we can estimate the kernel is certain conditions such as the case of turbulence when empirical formulas for making blur kernel is known. Getting blur kernel model and its parameter values is possible by experimenting with these techniques. In general, the direct division of blurred image by kernel in frequency domain is unstable. The problem also becomes significantly harder as the noise increases. For our experiment we use the model as describe above and mathematically expressed below. Here G, H, F and N are blurred image, blur kernel, original image and noise respectively in the frequency domain.

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

In spatial domain, this can be expressed as below. Here g, h, f and η are the blurred image, blur kernel, original image and noise respectively in the spatial domain and $\star$ denotes convolution.

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

In this report, we get the restored image using above mentioned techniques and compare them using peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics for different values of parameters. We also explore how the performance degrades as the magnitude of noise is increased.

II. BACKGROUND AND RELATED WORK

A. Full Inverse and Truncated Inverse Filtering

Full inverse of a blur kernel is taken by direct division of blurred image by blur kernel in the frequency domain. If the noise magnitude is small, we can expect to get a fairly good approximation of the original image. Though this is not generally the case because these images have a finite precision and division by small values in the blur kernel in frequency domain is unstable. As we shall see later, disturbing artifacts are visible is restored image because of this.

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

To get around this problem, we truncate the blur kernel in frequency domain, replacing any value below a certain threshold by one. This ensures a stable division. As we shall see later, the results are significantly better.

B. Approximate Wiener Filtering

The expression given below for this filter is obtained by minimizing the means squared error between the predicted and original image and assuming the degradation model assumed earlier.

$$\hat{F}(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \cdot G(u,v)$$

The parameter K, which is the ratio of power spectrum of noise and power spectrum of original image, is unknown and can be determined by observing the improvement in image as it is varied.

C. Constrained Least Squares Filtering

The objective function for CLS filtering is different from Wiener filtering. Here, we minimize the Laplacian of the restored image which is a measure of smoothness of the image. This minimization is done with the constraint of the degradation model we assumed. The result of this minimization is given below.

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

Here $P(u,v)$ is the Laplacian kernel is the frequency domain. Paramter γ is unknown and is varied till we get the best results.

D. Peak Signal-to-noise Ratio (PSNR)

The expression for $PSNR$ is given below. Here L is the maximum possible difference between 2 pixels (example: $L = 255$ for 8 bits) and MSE is the mean-squared error between the two images.

$$PSNR = 10 \cdot \log_{10} \left(\frac{L^2}{MSE} \right)$$

*Roll number 15D070022, Dual Degree 2015-2020, IIT Bombay
*Code can be found on GitHub.

E. Structural Similarity (SSIM)

Expression for SSIM is given below. One can verify that SSIM varies between 0 and 1. SSIM$(x, y) = 1$ when the two images x and y are exactly same. Here μ_x, μ_y σ_x^2, σ_y^2 and σ_{xy} are the mean of x, mean of y, variance of x, variance of y and covariance of x and y.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

$c_1 = 0.01L^2$ and $c_2 = 0.03L^2$ are used to stabilize denominator. (L is same as the L in PSNR.)

III. Experiments and Results

We divide this part into four parts. In the first three parts, we explore the three techniques described earlier but we use zero noise in the image i.e. $\eta(x, y) = 0$ everywhere. In the fourth part, we explore how the performance of these three techniques changes and we increase the noise.

Original image is shown in Fig. 1. The four blur kernels are shown in Fig. 2. For reference, a blurred image is shown in Fig. 3

Fig. 1. Original image.

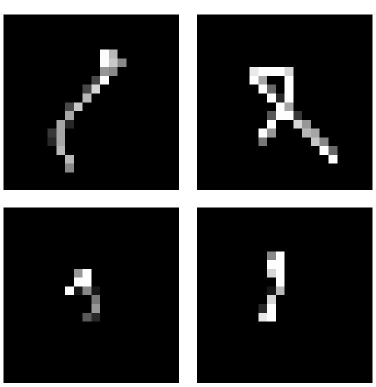

Fig. 2. Four blur kernels each with size 20x20 pixels.

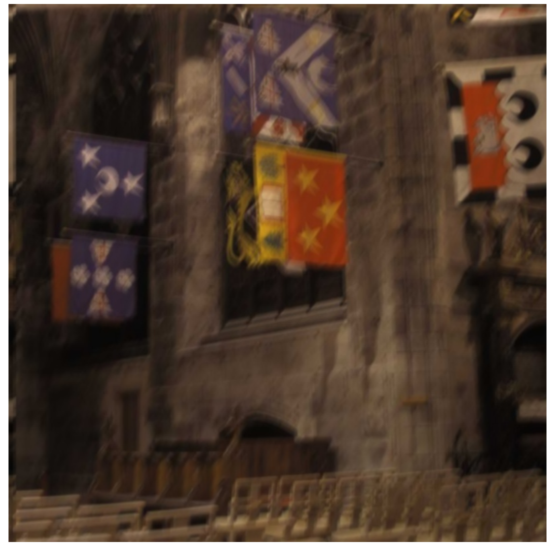

Fig. 3. Image blurred with first kernel.

A. Full Inverse and Truncated Inverse Filtering

Table I and Table II shows the PSNR and SSIM values respectively for different kernels. The threshold for truncation is shown in braces for the best results obtained.

TABLE I

PSNR values for different kernels for full inverse and truncated inverse filtering.

Kernel/Image	Blurred	Full Inv.	Truncated
1	20.93	25.32	35.15 (0.1)
2	20.65	25.01	33.98 (0.1)
3	20.29	28.34	36.34 (0.1)
4	20.67	27.12	36.06 (0.1)

TABLE II

SSIM values for different kernels for full inverse and truncated inverse filtering.

Kernel/Image	Blurred	Full Inv.	Truncated
1	0.745	0.921	0.991 (0.1)
2	0.734	0.916	0.989 (0.1)
3	0.722	0.959	0.993 (0.1)
4	0.742	0.947	0.993 (0.1)

We observe that both full inverse and truncated inverse give a gain in PSNR and SSIM over the blurred image but truncated inverse is much better than the full inverse. Outputs of full inverse filtering and truncated inverse filtering for first kernel are shown in Fig. 4 and Fig. 5 respectively. On close inspection of output of full inverse, we see that output has very high frequency artifacts though the blur clearly gone. The output of truncated inverse has no such artifacts. This observation can be attributed to the low values of blur kernel in frequency domain because of which direct division is unstable. Later, we shall see how the noise added to the image affects the output of this filtering.

Fig. 4. Output of full inverse when image is blurred with first kernel.

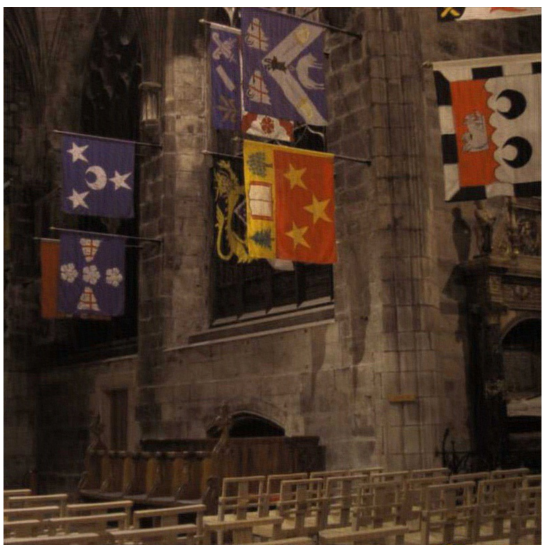

Fig. 5. Output of truncated inverse when image is blurred with first kernel.

B. Approximate Wiener Filtering

Table III and Table IV shows the PSNR and SSIM values respectively for different kernels. The K value for Weiner filter is shown in braces for the best results obtained.

TABLE III
PSNR VALUES FOR DIFFERENT KERNELS FOR WIENER FILTERING.

Kernel/Image	Blurred	Filtered
1	20.93	36.63 (0.01)
2	20.65	35.13 (0.01)
3	20.29	37.43 (0.01)
4	20.67	36.61 (0.01)

We observe that output of Weiner filter gives a significant gain in PSNR and SSIM over the blurred image. Further, the PSNR and SSIM values of output from Weiner filter is consistently better than output from full and truncated inverse filtering. Outputs of Weiner filtering for first kernel are shown in Fig. 6.

TABLE IV
SSIM VALUES FOR DIFFERENT KERNELS FOR WIENER FILTERING.

Kernel/Image	Blurred	Filtered
1	20.93	0.992 (0.01)
2	20.65	0.991 (0.01)
3	20.29	0.995 (0.01)
4	20.67	0.994 (0.01)

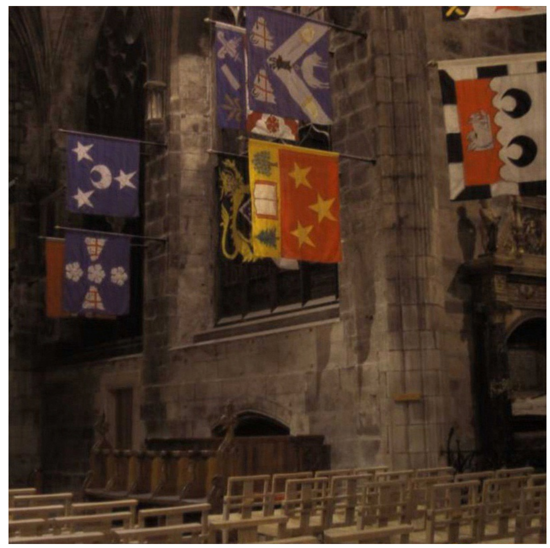

Fig. 6. Output of Wiener filtering when image is blurred with first kernel.

C. Constrained Least Squares (CLS) Filtering

Table V and Table VI shows the PSNR and SSIM values respectively for different kernels. The γ value for CLS filter is shown in braces for the best results obtained.

TABLE V
PSNR VALUES FOR DIFFERENT KERNELS FOR CONTRAINED LEAST SQUARED FILTERING.

Kernel/Image	Blurred	Filtered
1	20.93	37.59 (0.01)
2	20.65	36.99 (0.01)
3	20.29	40.34 (0.01)
4	20.67	39.00 (0.01)

We observe that output of CLS filter gives a significant gain in PSNR and SSIM over the blurred image. Further, the PSNR and SSIM values of output from CLS filter is consistently better than output from full and truncated inverse filtering as well as Weiner filtering. Outputs of CLS filtering for first kernel are shown in Fig. 7.

TABLE VI

SSIM VALUES FOR DIFFERENT KERNELS FOR CONTRAINED LEAST SQUARED FILTERING.

Kernel/Image	Blurred	Filtered
1	20.93	0.995 (0.01)
2	20.65	0.994 (0.01)
3	20.29	0.997 (0.01)
4	20.67	0.996 (0.01)

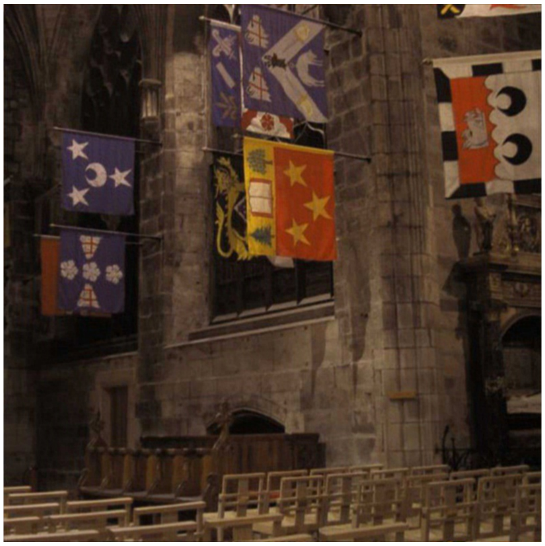

Fig. 7. Output of CLS filtering when image is blurred with first kernel.

To compare all the filtering techniques the combined tables for PSNR and SSIM are shown in Fig. VII and Fig. VIII respectively.

TABLE VII

COMBINED TABLE OF PSNR VALUES FOR DIFFERENT FILTERS.

Kernel	Blurred	Full Inv.	Truncated	Weiner	CLS
1	20.93	25.32	35.15	36.63	37.59
2	20.65	25.01	33.98	35.13	36.99
3	20.29	28.34	36.34	37.43	40.34
4	20.67	27.12	36.06	36.61	39.00

TABLE VIII

COMBINED TABLE OF SSIM VALUES FOR DIFFERENT FILTERS.

Kernel	Blurred	Full Inv.	Truncated	Weiner	CLS
1	0.745	0.921	0.991	0.992	0.995
2	0.734	0.916	0.989	0.991	0.994
3	0.722	0.959	0.993	0.995	0.997
4	0.742	0.947	0.993	0.994	0.996

D. Noise

To observe how noise affects, we add noise to the blurred image before filtering it. For this, blurred image corresponding to the first kernel is taken and zero-mean Gaussian noise with variance varying from 10^{-2} to 10^{4} is added to it. The plot of the PSNR and SSIM values of the recovered image using CLS filtering relative to the ground truth is show in Fig. 8. The plot confirms that increase in noise adversely affects the performance of filter. Similar plots are observed for other filters as well.

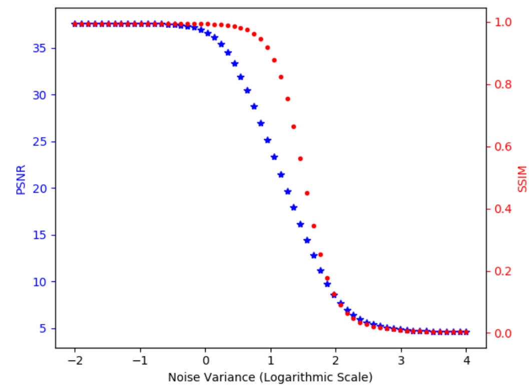

Fig. 8. Plot of PSNR and SSIM as the noise variance is varied from 10^{-2} to 10^{-4}. Left axis has labels for PSNR and right for SSIM.

Further, we can compare the magnitude of noise variance required to get, say 10dB PSNR, degradation in performance of different filters relative to best performance. Table IX shows the variance of noise required to get 10dB degradation in PSNR.

TABLE IX

NOISE VARIANCE REQUIRED TO GET 10DB DEGRADATION IN PSNR PERFORMANCE OF FILTER RELATIVE TO BEST PERFORMANCE.

Filter	σ_N^2
Full Inverse	2.1
Truncated Inverse	3.3
Wiener	3.4
CLS	6.7

IV. REAL-LIFE EXAMPLE

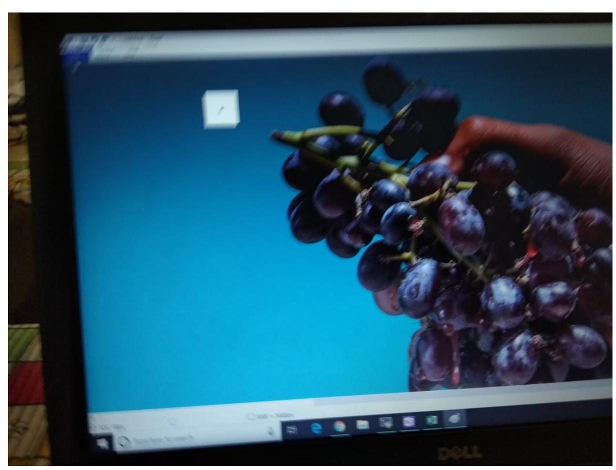

Fig. 9. Image taken from using phone camera of a laptop screen with motion during the click.

To derive the kernel, I took the part of image the part of the where there is a black dot in Fig. 9. On processing this part by inversion, followed by thresholding and increasing the dynamic range, I obtained the kernel in Fig. 10. The output using this kernel is shown in Fig. 11. Some improvement in image is clearly visible as the intensity of the dot has increased and the wrinkles on hand are more clearly visible after deblurring.

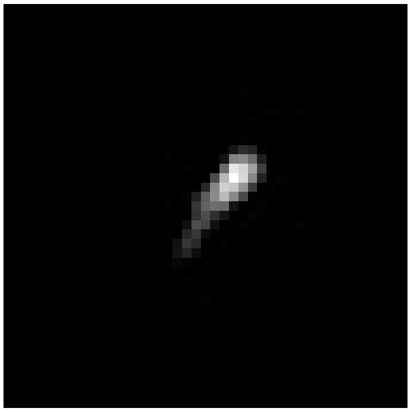

Fig. 10. Derived kernel from the motion of dot in the image of laptop screen. Preprocessing used includes taking mean across 3 channels, thresholding, increasing dynamic range and inversion. Size is 43x43.

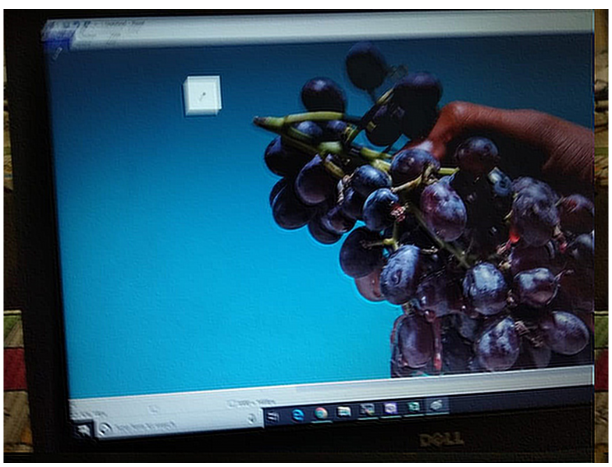

Fig. 11. Output of the constrained least square filtering with $\gamma = 0.02$ using the derived kernel.

V. DISCUSSION AND CONCLUSIONS

From Table VII and Table VIII, we conclude that CLS filter performs better than Wiener filter which is better than truncated inverse filter. Also, the metrics used, PSNR and SSIM, are consistent with each other in the sense that best results for both metrics happen to occur at the same parameter values and don't contradict. Because the kernels are very similar in terms of spatial extent, we also observed the best performance at same parameter values for all of them. Further, from Table IX, we can say that CLS filter is very robust to noise as compared to other filters.

REFERENCES

[1] Digital Image Processing, Gonzalez and Woods, Prentice Hall.
[2] Z. Wang, E. P. Simoncelli, and A. C. Bovik, Multi-scale structural similarity for image quality assessment, Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1398-1402, Nov. 2003.
[3] Python Official Documentation
[4] NumPy Official Documentation
[5] Matplotlib Official Documentation