Assignment 2: Image Restoration

A Report by

Sumit Kumar

IIT Bombay
183074017@iitb.ac.in

***Abstract*—In today's world *Image processing* is being used in almost all fields of technology whether it's space, military or medical science. Hence it is imperative to understand the basic concepts and key techniques in processing an image. One of the widely used application of Image Processing is Image restoration or Image enhancement. This paper describes some of the basic Image restoration techniques by the help of an image editor application which was designed and implemented as a part of this assignment**

Keywords*—*Image restoration, image transformations

I. Introduction

This report describes the design and development of an image processing application, various restoration mechanisms applied on different images, their results, the learning and conclusions. One of the key areas of image restoration is in the field of de-blurring a blurred image. An image can be blurred due to various reasons. It may be due to over or under exposure of the subject, or may be due to faulty sensor. Many a times there is motion blur in an image due to relative motion between the subject and camera. Apart from reasons of one's personal interest, image de-blurring plays important role in the area of medical science, space, defense, forensics etc. With the aim of gaining some basic hindsight in this area of image processing along with hands-on experience, this exercise was carried out, which this report describes.

This report is organized as follows. Section II describes the background study and related work. It also describes the various approaches that has been carried out in this exercise. Section III describes the results obtained on various approaches and compares them. The various learnings and conclusions have been summaries in Section IV.

II. Background read & Approach

A. Study

Some restoration techniques are best formulated in the spatial domain, while others are better suited for the frequency domain [1]. It may be noted here that, there is subtle difference between image enhancement and image restoration. While image enhancement is subjective, differs amongst viewers, image restoration is objective and has quantified parameters defined for it.

If an image is corrupted by additive noise, spatial domain restoration methods are suitable. But if the image is degraded by blur, frequency domain filtering techniques are better suited. The key concept in image restoration process is knowledge of the image degradation process i.e. how the image was degraded. Once, the degradation process or model is known, image restoration can become trivial. However, in most practical scenarios, image degradation model is not available to work with. So, approximation or prediction of such a degradation model becomes essential part of image restoration. This is the prime reason why most of the restoration mechanisms are interactive, as they tend to predict the degradation model.

There are various models of noise. Some important models amongst them are Additive Gaussian noise, Rayleigh, Gamma, Exponential, Uniform, Impulse or Salt & Pepper noise. The noise can be randomly distributed or uniformly distributed or can be periodic in nature. Spatial filtering such as mean filters or median filters can be used to restore images in such scenarios. Also, based on the local characteristics of an image, adaptive filtering methodology can be applied for image restoration.

If the image degradation is due to blur, as discussed earlier, the restoration processes are mostly employed in frequency domain. Some of the commonly used methods are Inverse Filtering, Minimum Mean Square Error (Wiener) Filtering, Constrained Least Square Filtering, and Geometric Mean Filtering. The efficiency of each of the above methods depends on the degradation model and is mostly an interactive process.

If the blurring kernel or degradation model is known, one of the simplest approach is Inverse Filtering. Seemingly looking simple, this approach may not always recover an image. This is because inverse filtering may amplify the high frequency noise, thus, making the image worse instead of enhancing it. One approach to get around the zero or small-value problem is to lime the filter frequencies to values near the origin [1].

In practical scenarios, mostly the degradation model is not known. The concept of machine learning to restore images in such scenarios can be handy. Recently, learning based methods have been used to achieve more efficient blur removal. A cognitive engine is used to *learn* the various relationship between the features of an input image. The learning phase is also called as training phase where large set of known input and output data set is used. The cognitive engine can be Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN). The efficiency of the machine learning based approach depends upon the type of cognition engine used, training data set used and also the training features used.

B. Approach

As a part of this exercise, Full-inverse filtering, Truncated Inverse filtering, Weiner filtering, and Constrained least square filtering techniques have been used to de-blur a given blurred image. Inverse filtering has been used when a part of blurring kernel is known. The de-blurring results can be significantly improved by truncating the inverse filter. The results of these approaches are described in next section.

C. Experimental Setup

A GUI application was developed to perform this exercise. User can select different blurred images, and can apply various de-blurring techniques. The de-blurring techniques were applied on the three color channels Red, Green, Blue of the image. User can control certain de-blurring parameters like K (Weiner Filtering), Y (Constrained Least Square Filtering), R (Truncated Filtering).

For developing the GUI application, open source and platform independent tool was a preferred design choice. QT is a widely used GUI tool. Most of the GUI tasks like opening file, taking user input become easier with QT. So PyQT4 was decided as the design choice for GUI development.

III. Experiments and Results

The image de-blur application was tested with various images. These images, along with de-blurring kernels were taken from [2]. The different subsections describe the various de-blur processes applied. Fig. 1 shows the original image (without any blur) along with de-blur kernels. Fig. 2. shows four different blurred images which we will try to de-blur in this exercise.

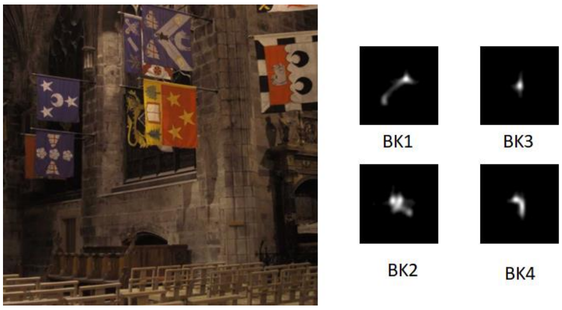

Fig 1. Original Image along with De-Blur Kernels

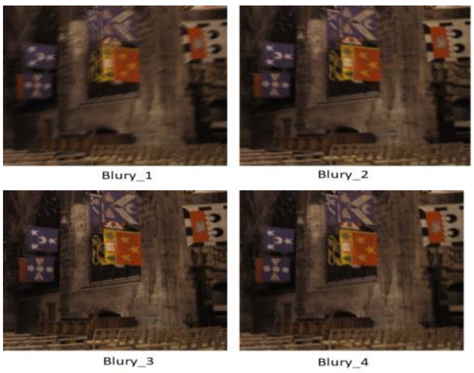

Fig. 2. The input blurry images

A. Full Inverse Filtering

This is one of the straight forward de-blur method. The de-blurred image is obtained by dividing the *blurred image* with the *de-blurring kernel* in frequency domain. The major drawback of this technique is that the de-blur kernel should be known *apriori*. Also, this technique magnifies the high frequency noise present in the blurred image. Fig. 3a. shows the results of full inverse filtering. As can be easily seen, in this case the full inverse filtering worsens the image, and the output gives almost no information about the input image. Fig 3(b) shows that if the PSF function is known, we can get back the reference image from Full Inverse.

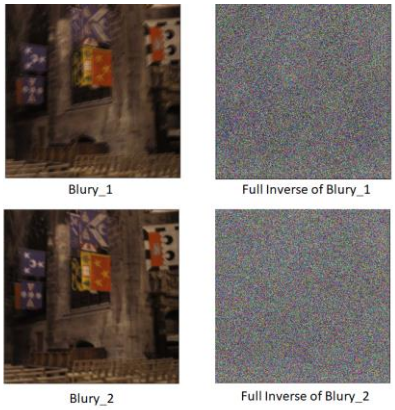

Fig. 3a. Output of Full Inverse Filtering when de-blur kernel is not known

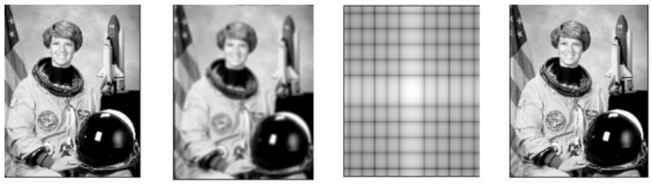

Fig. 3b. Output of Full Inverse Filtering when De-Blur kernel is known

B. Truncated Filtering

The central idea in truncated filtering is that we divide the blurred image (in frequency domain) by the de-blurring kernel not for the entire frequency range but within a radius. Outside the radius, the image will be unchanged. This prevents from magnifying high frequency noise. The result of truncated filtering for various radius and input images is shown in Fig. 4. The input image is Blury_1. As the radius increases, the output resembles full inverse filtering.

Fig. 4. Truncated Inverse filtering with radius 0, 1, 25, 50 (top row) 100, 225, 250, 400 (bottom row)

C. Wiener Filtering

Much improved results are obtained if wiener filtering is applied. The wiener filtering takes into account the factor 'K', which is the signal to noise ratio. The output of wiener filtering is shown in Fig. 5.

D. Constrained Least Square Filtering

Another common mechanism of de-blurring an image is Constrained Least Square Filtering. In this method, a Laplacian operator is included during the de-blur process. The result of applying this mechanism is shown in Fig. 6.

Fig. 5(a)

Fig. 5(b)

Fig. 5(c)

Fig. 5(d)

Fig. 5. Weiner De-blurring for different blurry images

Fig. 6(a)

Fig. 6(b)

Fig. 6(c)

Fig. 6(c)

Fig. 6. Constrained Least Square De-blurring for different blurry images

Fig. 5. shows the results obtained by applying Wiener Filtering. The first image in each row is the input blurry which correspond to input images of Fig. 2. Each input blurry image is de-blurred using corresponding Point Spread Function (PSF) or de-blurring kernel, which corresponds to Fig. 1. The other three images in each row is the resulting image after application of wiener filtering with K values being 1, 10, and 20. It may be noted here that the PSF used in this case was not normalized.

Fig. 6. Shows the results obtained by applying Constrained Least Square Filtering. The first image in each row is the input images as shown in Fig.2 and the PSF used to de-blur them correspond to Fig. 1. The other three images in each row are the resulting images after application of least constrained square filtering for Y values 1, 10, and 20.

Table 1. Comparison of various de-blurring techniques

Technique	Remarks
Full Inverse Filtering	Works when the *exact* PSF is known
Truncated Inverse Filtering	Works only for smaller radius. For the given image, R around 100 was found best
Wiener Filtering	Performance depends on PSF
CLS Filtering	Output image varied very slowly with Y

IV. PSNR & SSIM

One of the metric of image comparison is Peak Signal to Noise Ratio. It gives a measure of the amount of noise present in a noisy or blurry image with respect to a clean image called *groundtruth.* Structural Similarity Index (SSMI) is another metric to measure the similarity between two images, assuming one is the reference image. Mean Square Error (MSE) metric gives the square error between any two images. As a part of this assignment, PSNR and SSIM are computed for all the output images. SSIM is computed by using compare_ssim() library of skimage. The observations for PSNR and SSIM for Blur_1image is shown in Table 2. The best values of R, K, Y are indicated in the bracket in PSNR column of each de-blur techniques The detailed PSNR, MSE, SSIM for each output image is contained in the output folder in the assignment repository. It should be noted that these values were calculated for *red channel* of the two comparing images. Although the absolute values may differ, but the relative values will be same if these metrics are calculated for a single channel or for all three channels combined for color images.

Table 2. Observation of PSNR & SSIM for Blur_1 image

Technique	PSNR (dB)	SSIM	Remarks
Full Inverse	< 0	0	Technique fails for unknown PSF
Truncated	12 (100)	0.55	PSNR slowly decreased with increasing radius
Wiener	66 (50)	0.76	PSNR increased with K, up to 50, it decreased after that
CLS	60.2 (100)	0.58	PSNR varies very slowly with change in Y

V. Debluring of Personal Image

One of the tasks of this assignment was to de-blur a personal image. Without knowledge of blurring kernel, de-blurring an image is a difficult problem. In most of the practical-scenarios, the modern technique to solve this problem is using machine learning. It helps us approximately predict the de-blurring kernel and improve an image. For the current assignment, a blurred image was taken. Various techniques were tried out for de-blurring. Firstly, a box-kernel was taken to de-blur, which wasn't quite effective. Different kernels, at random, were tried, and techniques like truncated filtering, Wiener filtering were applied with un-satisfactory results. However, some better results were obtained by Constrained Least Square filtering. Kernel (Blurry1_4_kernel) was taken and constrained least square filtering method was applied. The result is shown in image 7 for various values of Y. The best result was obtained for Y = 25.

Fig. 7. De-blurring an image without any ground-truth. 1: Original Blurry image; 2, 3, 4: De-blurred image with constrained least square filtering approach for Y = 2, 10, 25.

VI. Links

.

The project is uploaded in github. The link to the project is: *https://github.com/sumitiitb/EE_610_Homework_2*

V. Discussions and Conclusion

A. Learnings

The assignment, similar to the previous one, was a great learning exercise.

To begin with, working with color image was challenging. Splitting the image into various color channels, processing the individual channels and then merging them was new and fun. It was a deliberate effort to shorten the program code and all the three channels were processed together rather than separately.

Implementing the de-blur techniques was fun. The best part was seeing a better image emerge from a blurry image. The techniques studied in theory when matches with practical experiment is a great learning experience itself.

When working with color images, scaling of pixel values is an important consideration. Sometime, the PSF kernel needs to be normalized out while at other times, the images needs to be normalized.

Just to check if Full Inverse works at all, I implemented a full inverse for a known PSF. This PSF is known in its entirety i.e. for the complete image. The result of full inverse, when it worked was nice to see.

At the end of this exercise, I feel I am more confident with python and handling of images in general.

B. Current Direction in Image Restoration

The contemporary techniques in image restoration or image enhancement is mostly focused on using Machine Learning techniques. With the spread of image based application in various fields of life from medicine, to defense and military to image recognition for authentication purposes, the technology for image enhancement is ever growing. Up-scaling of image and video files at the user end and down-scaling of image or video at transmission end is ever growing need and so it's an ever-green area of research..

Acknowledgment

Firstly, I would like to acknowledge my professor Prof. Amit Sethi for making me learn the concept of image processing through this practical assignment. I would also like to acknowledge the help I received from the online tutorials and videos from various channels across the web. I would also like to acknowledge my friends who helped me learn some of the tricks of python.

References

[1] Digital Image Processing, Third Edition, by R.C Gonzalez and R.E Woods : For the basic concepts of Image processing

[2] Class Notes: For the image blur through increasing kernel window size

[3] http://webdav.is.mpg.de/pixel/benchmark4camerashake/

[4] Python Tutorial : PyQt with Python GUI Programming tutorial (https://www.youtube.com/channel/UCfzlCWGWYyIQ0aLC5w48gBQ)

[5] Wikipedia : For understanding the various implementation algorithms

[6] Stackoverflow (www.stackoverflow.com) : for correcting syntax errors !

[7] Google (www.google.com): for practically everyting !

[8] https://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/

[9] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py transforms/py_fourier_transform/py_fourier_transform.html

[10] https://stackoverflow.com/questions/19739503/dft-matrix-in-python

[11] https://www.pyimagesearch.com/2014/09/15/python-compare-two-images