

Implementation of Image Restoration Techniques in MATLAB

Jitendra Suthar¹, Rajendra Purohit²
Research Scholar¹, Associate Professor²
Department of Computer Science,
JIET, Jodhpur

Abstract:- Processing of digital images with the help of digital computers known as Digital Image Processing. One of the most applicable areas in Image Processing methods is to enhance the pictorial information for human perception. Image restoration is a method to clearing the degraded image to obtain the original image. For years researchers have been working in developing new techniques that can restore the original image from degraded image. The aim of this paper is to demonstrate the implementation of different types of techniques for image Restorations in MATLAB. MATLAB is very powerful tool for image processing because it support all types of image format and conversion between them. It also support all types of datatypes.

Index terms: Blur, MATLAB, Image processing tool, Deblurring, PSF and Image Restoration.

I. INTRODUCTION

Image restoration is an old problem in the field of image processing, one that continues to accumulate attention from academics and businesses alike [8]. Its application areas are observed in many different real-world problems and work as an easy way to visualize examples of a larger range of inverse problems in many fields [1]. Image Processing Tool provides a complete set of standard algorithms and graphical tools for image processing, visualization, analysis and algorithm development. It able to restore noisy or distorted images, analyze shapes and textures in image, enhance the viewing quality of images, extract desired features from image and also register two images. Image Processing Tool provide facility to scientists and engineers in areas such as biometrics, semiconductor testing, surveillance, gene expression, microscopy, remote sensing, image sensor design, materials science and colour science.[2]

II. INTRODUCTION TO MATLAB

Most functions in Image Processing Tool are written in the open source MATLAB language, giving us the facility to inspect the algorithms, create our own custom functions and modify the source code.

A. Key Feature

- Image smoothing, image sharpness, image enhancement, deblurring, contrast enhancement and filters design.
- Image analysis, including feature detection, segmentation, morphology and measurement.
- Image registration and spatial transformations.
- Image transforms, including, DCT, DWT, FFT, IFFT, Radon and fan-beam projection.
- DICOM (Digital Imaging and Communications in Medicine) file import and export.

B. Image Formats in MATLAB

There is lossy and lossless algorithm. Lossless algorithms will reduces size without degrading, quality of picture so it cannot compress file to a very small size for better compression. Lossy method allows reduces image size but it will degrade the quality of picture. It able to analyze, visualize and process these images in different data structure (data types), including floating point (single precision and double precision) representation and signed or unsigned 8 bit, 16 bit and 32 bit integer values. There are many ways to process an image in MATLAB

environment for import and export purpose. File formats can also be divided according to type of graphics. In Raster the most commonly used formats are GIF, JPEG, BMP, PGM and PNG. But in vector graphic there is lack of formats available for use because most of them are specified by an image processing program, like for example CorelDraw (.cdr) or AutoCAD (.dwg). EPS (Encapsulated PostScript) format is worth noticing. It can be implemented to save both types of graphics.

C. Conversion between different formats

Table 1: Image Format Conversion

Operation	Command
Convert intensity formats of image to indexed format.	gray2ind()
Convert indexed/ intensity /RGB formats to binary format.	dither()
Convert indexed formats of image to intensity format.	ind2gray()
Conversion of a regular matrix to intensity format by scaling.	mat2gray()
Convert indexed formats to RGB format.	ind2rgb()
Convert RGB formats of image to indexed format.	rgb2ind()
Convert RGB formats of image to intensity format.	rgb2gray()

D. Reading files and Work with variables

In real life when we encounter an image we want to work with, it is generally in form of a file. Once after processing, it may require to write it back to a JPEG-file or in other File format. This is done using the imread and imwrite MATLAB commands.

Table 2: Command for read and write an image

Operation	Command
To read an image in Matlab environment.	imread()
To write an image to a file form Matlab environment.	imwrite(,)

Once it has read a file, it is automatically converted into an intensity format of image (a matrix). Once we are done calculation, we may want to restore the matrix representing the image in order to continue to work with this matrix in future. This is done by using the commands load and save.

III. IMAGE RESTORATION TECHNIQUES

This section assumes that PSF is known prior to restoration.

1) Wiener Filtering

Wiener filtering is most useful technique for image deblurring. The wiener filter has a good performance to reduce the blur effect in images caused by unfocussed optics or linear motion. In real life the corrupted image is a result of poor sampling. Usually individual pixel in the image should consist intensity value for a single stationary point in front of the capturing device like as camera. Unfortunately, if the shutter speed is very slow or camera is moved, a given pixel will be a mixture of intensities from points along the line of the camera's motion [3, 4].

Algorithm

1. Performing some preprocessing by using Median Filter before restoring the blurred image.

- ```

I = medfilt2 (abs (I));
2. Convert the input image into frequency domain.
 I_fft = fft2 (I);
3. Design the Point Spread Function (PSF) for degradation of image.(For example type of blur is motion blur)
 Psf = fspecial ('motion', Length, Theta);
4. The size of Psf is not equal to image size so convert Psf to Otf (Optical Transfer Function)
 of desired size.
 Otf = psf2otf (Psf, size (I_fft));
5. Calculate the Conjugate value of Otf for modulus acquisition
 Otfc = conj (Otf); Mod_Otf =
 Otf.*Otfc;
6. Perform some processing to avoid divide by zero error in image processing for x = 1: size(Otf, 1)
 for y = 1: size(Otf, 2)
 if Otf (x, y) == 0
 Otf (x, y) = 0.000001;
 end
 end
end end
7. Deblurring input blurred image by using WEINER FILTER formula
 Debl = ((Mod_Otf. / (Mod_Otf + SNR)). / (Otf)) .* I_fft; I_debl = ifft2 (Debl);
8. Restored image
 Re_image = I_debl;

```

**Flowchart:**

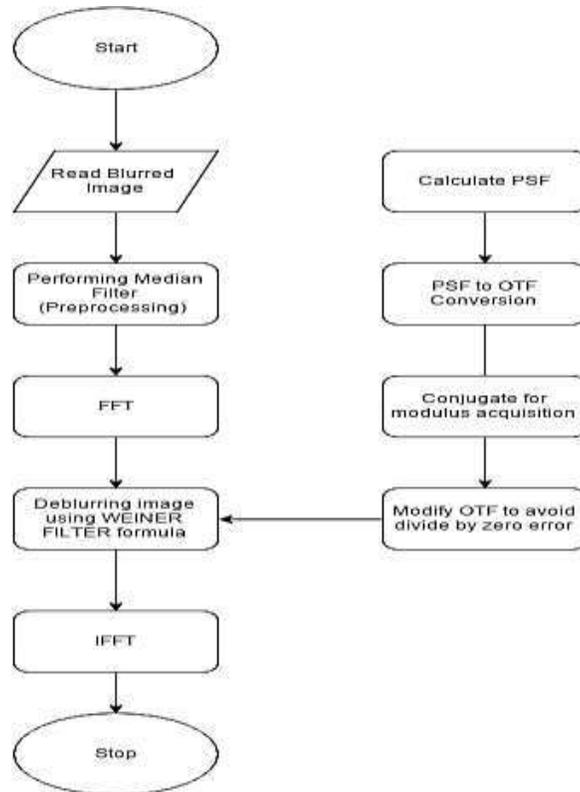


Figure 1: Flow chart for Wiener Filter

## 2) Lucy-Richardson Deconvolution

This algorithm has been independently proposed by Lucy [5] and Richardson [6]. The L-R algorithm is an iterative technique that maximizes a Poisson statistics image model likelihood function. [7]

### Algorithm:

1. Performing some preprocessing  
 $I = \text{medfilt2}(\text{abs}(I));$
2. Initializing the initial estimation to the blurred image  
 $\text{Est} = I;$
3. Design the Point Spread Function (PSF) for degradation of image.(For example type of blur is motion blur)  
 $\text{Psf} = \text{fspecial}('motion', \text{Length}, \text{Theta});$
4. The size of Psf is not equal to image size so convert Psf to Otf (Optical Transfer Function) of desired size.  
 $\text{Otf} = \text{psf2otf}(\text{Psf}, \text{size}(I_{\text{fft}}));$
5. Repeat step 6 to 13 number of iteration is given by user.
6. Convert the estimate to frequency domain by using FFT from spatial domain.  
 $F_{\text{est}} = \text{fft2}(\text{Est});$
7. Multiply the Otf with the estimate in frequency domain (step 6).  
 $F_{\text{lest}} = \text{Otf} .* F_{\text{est}};$
8. Converting back to the blurred image estimate to spatial domain by using IFFT.  
 $I_{F_{\text{lest}}} = \text{ifft2}(F_{\text{lest}});$
9. Calculating new value by taking ratio of blurred image and estimate of the deblurred image.  
 $I_{\text{ratio}} = I ./ I_{F_{\text{lest}}};$
10. Converting the ratio to frequency domain by spatial domain by using FFT.  
 $F_{\text{iratio}} = \text{fft2}(I_{\text{ratio}});$
11. Calculate the correction vector for deblurring.  
 $\text{Corr\_vec} = \text{Otf} .* F_{\text{iratio}};$
12. Converting the correction vector to spatial domain from frequency domain by using IFFT.  
 $I_{\text{Corr\_vec}} = \text{ifft2}(\text{Corr\_vec});$
13. To find new estimate multiply the calculated correction vector with the estimate of deblurred image.  
 $\text{After\_corr} = I_{\text{Corr\_vec}} .* \text{Est};$   
 $\text{Est} = \text{After\_corr};$



Figure 2: Flowchart for Lucy Method

#### IV. RESULTS

For testing total 4 image are used and they regular captured images, in which first two are naturally blurred image and represented as R\_1, R\_2 and further two images are manually blurred represented as R\_3, R\_4 .

Table 3: Input images

| S.no | Type of Blur | Name | Size    |
|------|--------------|------|---------|
| 1.   | Naturally    | R_1  | 180*180 |
| 2.   | Naturally    | R_2  | 361*480 |
| 3.   | Manually     | R_3  | 300*300 |
| 4.   | Manually     | R_4  | 240*320 |

Table 4: Results of input images

| S | Input   | Size    | Value | Wiener | Lucy  |
|---|---------|---------|-------|--------|-------|
| 1 | R_1.jpg | 180*180 | PSNR  | 39.98  | 48.43 |
|   |         |         | RMSE  | 0.01   | 0.003 |
| 2 | R_2.jpg | 361*480 | PSNR  | 35.49  | 36.16 |
|   |         |         | RMSE  | 0.014  | 0.013 |
| 3 | R_3.bmp | 300*300 | PSNR  | 37.23  | 43.28 |
|   |         |         | RMSE  | 0.010  | 0.005 |
| 4 | R_4.bmp | 240*320 | PSNR  | 35.61  | 52.20 |
|   |         |         | RMSE  | 0.016  | 0.002 |

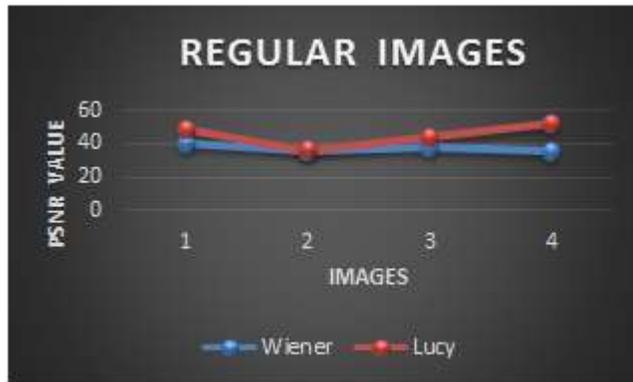


Figure 3: PSNR values of Wiener, Lucy and Novel method for Regular images

**Output:**

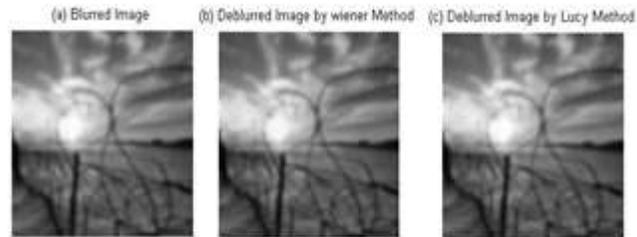


Figure 4: Output of R\_1 image



Figure 5: Output of R\_2 image



Figure 6: Output of R\_3 image



Figure 7: Output of R\_4 image

In all above figure label „a“ shows the Original image and label „b“ shows the deblurred by Wiener filter and label „c“ shows the deblurring by Lucy method.

## V. CONCLUSION

So Lucy-Richardson algorithm is the best algorithm for deblurring of blurred images in non-blind techniques. It is clear from table number 4 that Lucy method provides better result as compare to Wiener filter because it produce higher PSNR and lower RMSE value.

## REFERENCES

1. S. Reeves, Image Deblurring – Introduction. 2007, “<http://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/>”.
2. Introduction to MATLAB referred link is “<http://in.mathworks.com/help/images/getting-started-with-image-processing-toolbox.html>”
3. P. Singh and L. Jain, "Noise reduction in ultrasound images using wavelet and spatial filtering techniques." In Information Management in the Knowledge Economy (IMKE), 2013 2nd International Conference on, pp. 57-63. IEEE, 2013.
4. A. Dwivedi and A. Singh, "An Advanced Filter for Image Enhancement and Restoration." Journal Open Journal of Advanced Engineering Techniques OJAET Vol 1, no. 1 PP. 7-10, 2013.
5. L. B. Lucy, “An iterative technique for the rectification of observed distributions”, *Astronomical Journal*, 79(6):745–754, 1974.
6. W. H. Richardson, “Bayesian-based iterative method of image restoration”, *Journal of Optical Society of America*, 62(1):55–59, January 1972.
7. Ratnakar Dash, “Parameters Estimation for Image Restoration”, PhD Thesis, National Institute of Technology Rourkela, March 2012.
8. Steffen Losch “Depth from Blur”, Bachelor’s Thesis, Saarland University, July 2009.