International Journal of Applied Engineering and Technology ISSN: 2277-212X (Online)
An Open Access, Online International Journal Available at http://www.cibtech.org/jet.htm
2014 Vol. 4 (2) April-June, pp.115-124/Saini and Himral
Research Article

IMAGE PROCESSING USING BLIND DECONVOLUTION DEBLURRING TECHNIQUE

***Sonia Saini[1] and Lalit Himral[2]**
[1]CSE Department, Kurukshetra University Kurukshetra, Haryana, India
[2]Yamuna Group of Institution, Yamunanagar- Haryana, India
**Author for Correspondence*

ABSTRACT
Image processing is an important component of modern technologies because human depends so much on the visual information than the creatures. The challenge to scientists, engineers and business people is to quickly extract the valuable information from the raw image data. This paper focused on image restoration which is sometimes referred to image deblurring or image deconvolution. Image restoration is concerned with the reconstruction or estimation of blur parameters of the uncorrupted image from a blurred and noisy one. The goal of blur identification is to estimate the attributes of the imperfect imaging system from the observed degraded image itself prior to the restoration process (Kundur and Hatzinakos, 1996). Blind Deconvolution algorithm can be used effectively when no information about the blurring and noise is known. The algorithm restores the image and the point spread function (PSF). The aim of this paper to show the effective Blind Deconvolution algorithm for image restoration which is the recovery in the form of a sharp version of blurred image when the blur kernel is unknown.

Keywords: *Blind Deconvolution, Image Recovery, PSF, Damped Lucy-Richardson Algorithm, MATLAB, Digital Image, Blurred Image*

INTRODUCTION
Today, the medical industry, astronomy, physics, chemistry, forensics, remote sensing, manufacturing, and defense are just some of the many fields that rely upon images to store, display, and provide information about the world around us. The challenge to scientists, engineers and business people is to quickly extract valuable information from raw image data. This is the primary objective of image processing i.e. converting images to information. Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with images. These include ultrasound, electron microscopy, and computer-generated images. Thus, digital image processing encompasses a wide and varied field of applications. Image processing is an important component of modern technologies because human depends so much on the visual information than other creatures. Image is better than any other information form for us to perceive. Among our information about the world, 99% is perceived with our eyes (Russ, 1995). Image processing has traditionally been an area in the engineering community. In the past few decades several advanced mathematical approaches have been introduced into this field, namely, the variational calculus, partial differential equations (PDE) and stochastic approaches (or statistical methods), and they have become important tools for theoretical image processing. Digital image processing is a subset of the electronic domain wherein the image is converted to an array of small integers, called *pixels*, representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer or other digital hardware. There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, and high-level processes. Low-level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images. Mid-level processes on images involve tasks such as segmentation (partitioning

an image into regions or objects), description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (e.g., edges, contours, and the identity of individual objects). Finally, higher-level processing involves making sense of an ensemble of recognized objects, as in image analysis, and, at the far end of the continuum, performing the cognitive functions normally associated with human vision (Gonzalez and Woods, 2009).

Image restoration is concerned with the reconstruction or estimation of blur parameters of the uncorrupted image from a blurred and noisy one. Essentially, it tries to perform an operation on the image that is the inverse of the imperfections in the image formation system. In the process of image restoration, the characteristics of the degraded system and the noise are assumed to be known a priori. In practical situations, however, one may not be able to obtain this information directly from the image formation process. The goal of blur identification is to estimate the attributes of the imperfect imaging system from the observed degraded image itself prior to the restoration process (Kundur and Hatzinakos, 1996).

Literature Review

Digital image processing was first used in newspaper industry during 1920s. Introduction of the Bartlane cable picture transmission system in the early 1920s reduced the time required to transport a picture across the Atlantic from more than a week to less than three hours. Specialized printing equipment coded pictures for cable transmission and then reconstructing them at the receiving end. Figure 1.1 was transmitted in this way and reproduced on a telegraph printer fitted with typefaces simulating a halftone pattern. Some of the initial problems in improving the visual quality of these early digital pictures were related to the selection of printing procedures and the distribution of intensity levels. The printing method used to obtain Figure 1.1 was abandoned toward the end of 1921 in favor of a technique based on photographic reproduction made from tapes perforated at the telegraph receiving terminal. Figure 1.2 shows an image obtained using this method. The improvements over Figure 1.1 are evident, both in tonal quality and in resolution.

Figure 1.1: A digital picture produced in 1921 from a coded tape by a telegraph printer with special type faces.

Figure 1.2: A digital picture made in 1922 from a tape punched after the signals had crossed the Atlantic twice

The early Bartlane systems were capable of coding images in five distinct levels of gray. This capability was increased to 15 levels in 1929. Figure 1.3 is typical of the type of images that could be obtained using the 15-tone equipment. During this period, introduction of a system for developing a film plate via light beams that were modulated by the coded picture tape improved the reproduction process considerably (Gonzalez and Woods, 2009). Improvements on processing methods for transmitted digital pictures continued to be made during the next 35 years. However, it took the combined advents of large-scale digital computers and space program to bring into focus the potential of image processing concepts. Work on using computer techniques for improving images from a space probe began at the Jet Propulsion Laboratory (Pasadena, California) in 1964 when pictures of the moon transmitted by Ranger 7 were processed by a computer to correct various types of image distortion inherent in the on-board television camera. These techniques served as the basis for improved methods used to enhance and restore the images (Gonzalez and Woods, 2001)

Figure 1.3: Unretouched cable picture of Generals Pershing and Foch, transmitted in 1929 from London to New York by 15-tone equipment

Although the examples just cited involve digital images, they are not considered digital image processing results in the context of our definition because computers were not involved in their creation. Thus, the history of digital image processing is intimately tied to the development of the digital computer. In fact, digital images require so much storage and computational power that progress in the field of digital image processing has been dependent on the development of digital computers and of supporting technologies that include data storage, display.

Proposed Work

The proposed efforts have been utilized to develop a Blind Deconvolution Algorithm that produces useful results when processing degraded scenes. This technique will capitalize on the statistics of the blurry image and the refined image estimate, in an iterative approach to converge on the correct seeing parameter. Image restoration methods can be considered as direct techniques when their results are produced in a simple one step fashion. Equivalently, indirect techniques can be considered as those in which restoration results are obtained after a number of iterations. Known restoration techniques such as inverse filtering and Wiener Filtering can be considered as simple direct restoration techniques. The problem with such methods is that they require knowledge of the blur function that is point-spread function (PSF), which is, unfortunately, usually not available when dealing with image blurring. The goal of this work is to develop a Blind Deconvolution algorithm for image restoration which is the recovery in the form of a sharp version of a blurred image when the blur kernel is unknown. The fundamental task of image deblurring is to de-convolute the blurred/degraded image with the PSF that exactly describes the distortion. Firstly, the original image is degraded using the Degradation Model. It can be done by Gaussian Filter which is low-pass filter used to blur in image. In the edges of degraded image, the ringing effect due to high frequency drop-off can be detected using Edge detection methods. This ringing effect should be removed before restoration using edge trapping. After removing the ringing effect, Blind Deconvolution algorithm is applied to the blurred images. It is possible to renovate the original image without having specific knowledge of degradation filter, additive noise and image spectral density.

Steps of Blind Deconvolution Algorithm

The proposed Blind Deconvolution algorithm consists of following steps:

Step 1: Read in Images: Images to be deblurred are read into MATLAB environment.

Step 2: Simulate Blur: In this step a real life blur will be simulated using different types of filter.

Step 3: Restore the Blurred image using PSF of various sizes: This step involves the restoration of image using Blind Deconvolution Algorithm by trying PSFs of different sizes.

Step 4: Improving the Restoration: There some ringing in the image, restored in previous step. To avoid this we will exclude the pixels affected by the ringing.

SIMULATION & RESULT

SIMULATION

Implementation of Blind Deconvolution Algorithm

Step I: Read in Images

Images are read into MATLAB environment using function *imread*. The *imread* reads a JPEG true color image IIE into a three dimensional image array RGB of size 352x558, shown in Figure 2.1. As no path information is included in *imread*, so it reads the image from current working directory. Function *rgb2gray* converts the true color image IIE into a grayscale image, shown in Figure 4.2.

Name	**Size**	**Bytes**	**class**
RGB	352x558x3	589248	uint8
I	352x558	196416	uint8

Figure 2.1: Original Image (True Color)

Figure 2.2: Original Image (Grayscale

Step II: Simulate Blur

In this step a blur (e.g. due to camera/object motion or lack of focus) is simulated on the image, read in previous step, by creating a PSF of dimensions7x6, using Gaussian filter. Then the PSF is convolved (using *imfilter*) with image array I to produce the blurred image which is shown in Figure 2.3.

Name	Size	Bytes	Class
PSF	7x6	336	double
Blurred	352x558	196416	uint8

Figure 2.3: Gaussian Blurred Image

Figure 2.4: Deblurring with Undersized PSF

In convolution the value of an output pixel is computed as a weighted sum of neighbouring pixels. The matrix of weights is called the convolution kernel. Here PSF is the convolution kernel. Border elements can be calculated with the help of symmetric function which is explained below.

STEP-III: Restore the Blurred Image Using PSF of Various Sizes

As already explained in the introduction of the proposed algorithm, it recovers the sharp version of the blurred image when the blur kernel (PSF) is unknown. This unknown blur kernel (PSF) or its approximate can be found by trying various sizes of it to deconvolve the image.

Name	Size	Bytes	Class
UNDERPSF	3x2	48	double
J1	352x558	196416	uint8
P1	3x2	48	double

To determine the size of the PSF, examine the blurred image and measure the width of the blur (in pixels) around an obviously sharp object. Because the size of the PSF is more important than the values it contains, we can typically specify an array of 1's as initial PSF.

▪ In first iteration we specify an array of 1's of size less than the size of the PSF. It is stored as an array named UNDERPSF.

$$\text{UNDERPSF} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Now we restore the image with the blind deconvolution algorithm using the function *deconvblind*. Restored image is shown in Figure 2.4.
deconvblind function implements the blind deconvolution algorithm, which performs deblurring without knowledge of the PSF. When you call *deconvblind*, you pass as an argument, your initial guess at the PSF. The *deconvblind* function returns a restored PSF in addition to the restored image. It gives a new PSF and the restored image

$$P1 = \begin{bmatrix} 0.16638 & 0.1671 \\ 0.16658 & 0.16733 \\ 0.16595 & 0.16666 \end{bmatrix}$$

▪ The second restoration, J2 and P2, uses an array of ones, OVERPSF, for an initial PSF that is 4 pixels longer in each dimension than the true PSF.

Name	**Size**	**Bytes**	**Class**
OVERPSF	11X10	880	double
P2	11X10	880	double
J2	352X558	196416	uint8

Figure 2.5: Deblurring with Oversized PSF

In this step we creates an array of oversized PSF, i.e. the size of the PSF is larger than the size of actual PSF, using function *padarray*. This is explained below with the help of an example.
Example
B = padarray(A,[2 3],'replicate','both')

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \\ 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Again *deconvblind* restores the image by Blind Deconvolution algorithm. Figure 4.5 shows the restored image.

$$OVERPSF = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$P2 = \begin{bmatrix} 0.00849 & 0.00858 & 0.00887 & 0.00882 & 0.00896 & 0.00908 & 0.00862 & 0.00891 & 0.00879 & 0.00866 \\ 0.00858 & 0.00869 & 0.00882 & 0.00898 & 0.00913 & 0.00926 & 0.00918 & 0.00306 & 0.00892 & 0.00877 \\ 0.00866 & 0.00879 & 0.00895 & 0.00913 & 0.00930 & 0.00943 & 0.00934 & 0.00920 & 0.00903 & 0.00886 \\ 0.00875 & 0.00890 & 0.00908 & 0.00928 & 0.00947 & 0.00960 & 0.00950 & 0.00934 & 0.00915 & 0.00995 \\ 0.00883 & 0.00901 & 0.00920 & 0.00942 & 0.00962 & 0.00976 & 0.00967 & 0.00947 & 0.00926 & 0.00904 \\ 0.00892 & 0.00901 & 0.00931 & 0.00953 & 0.00974 & 0.00988 & 0.00956 & 0.00957 & 0.00935 & 0.00912 \\ 0.00883 & 0.00900 & 0.00920 & 0.00941 & 0.00965 & 0.00974 & 0.00963 & 0.00944 & 0.00923 & 0.00900 \\ 0.00874 & 0.00889 & 0.00907 & 0.00927 & 0.00945 & 0.00958 & 0.00946 & 0.00929 & 0.00909 & 0.00889 \\ 0.00865 & 0.00878 & 0.00894 & 0.00912 & 0.00928 & 0.00939 & 0.00929 & 0.00913 & 0.00895 & 0.00877 \\ 0.00857 & 0.00868 & 0.00882 & 0.00897 & 0.00911 & 0.00921 & 0.00912 & 0.00898 & 0.00883 & 0.00867 \\ 0.00850 & 0.00859 & 0.00871 & 0.00884 & 0.00896 & 0.00905 & 0.00896 & 0.00849 & 0.00872 & 0.00859 \end{bmatrix}$$

The third restoration, *J3* and *P3*, uses an array of ones, INITPSF, for an initial PSF that is exactly of the same size as the true PSF.

Name	**Size**	**Bytes**	**Class**
INITPSF	7x6	336	double
J3	352x558	196416	uint8
P3	7x6	336	double

$$P3 = \begin{bmatrix} 0.023625 & 0.023662 & 0.023733 & 0.023904 & 0.023854 & 0.023808 \\ 0.023675 & 0.023722 & 0.023802 & 0.023983 & 0.023926 & 0.02387 \\ 0.023705 & 0.023759 & 0.023848 & 0.024037 & 0.023976 & 0.023911 \\ 0.023797 & 0.023857 & 0.023953 & 0.024151 & 0.024083 & 0.024013 \\ 0.023642 & 0.023696 & 0.023786 & 0.023976 & 0.023914 & 0.023849 \\ 0.023575 & 0.02362 & 0.023703 & 0.023884 & 0.023826 & 0.023768 \\ 0.023546 & 0.023581 & 0.023656 & 0.023827 & 0.023775 & 0.023725 \end{bmatrix}$$

$$INITPSF = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figure 2.6: Deblurring with INITPSF

Figure 2.12: Deblurred Image (From Gaussian Blur

STEP IV: Improving the Restoration

The ringing in the restored image, *J3* (Figure 2.6), occurs along the areas of sharp intensity contrast in the image and along the image borders. This example shows how to reduce the ringing effect by specifying a weighting function. The algorithm weights each pixel according to the WEIGHT array while restoring the image and the PSF. In our example, we start by finding the "sharp" pixels using the *edge* function. By trial and error, we determine that a desirable threshold level is 0.34.

Name	**Size**	**Bytes**	**Class**
WEIGHT	352x558	1571328	double
Se	1x1	729	strel object

Although *deconvblind* was able to deblur the image to great extent, the ringing around the sharp intensity contrast areas in the restored image is unsatisfactory.

We repeat the deblurring process, attempting to achieve a better result by –

→ Eliminating high contrast areas from the processing,

→ Specifying a better PSF

The image is restored by calling *deconvblind* with the WEIGHT array and an increased number of iterations (35). Almost all the ringing is suppressed**.**

Effect of Changing the Filter to Simulate a Blur Using Average Filter

PSF generating element of average filter is

*h = ones(n(1),n(2)) /(n(1)*n(2))*

Let's take an example

Suppose

n(1) = 3

n(2) = 4

Then h will be an array of ones with dimensions 3x4.

$$h = \frac{1}{12}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$h = \begin{bmatrix} 0.0833 & 0.0833 & 0.0833 & 0.0833 \\ 0.0833 & 0.0833 & 0.0833 & 0.0833 \\ 0.0833 & 0.0833 & 0.0833 & 0.0833 \end{bmatrix}$$

Figure 2.13: Average Blurred Image

……Repeating All the Steps of Restoration Using Blind Deconvolution Algorithm……

Figure 2.14: Deblurred Image (From Average Blur

RESULTS

In this work, grayscale/two dimensional images are processed. The image size is chosen to be 352x558. Image was degraded with PSF using blurring filter of size 7x6 using different blur mechanisms like Gaussian and Average blur methods. To deblur these images the Blind Deconvolution algorithm is applied to determine the weight threshold and number of iterations to obtain the best quality of the restored image. The results obtained using Gaussian and Average blur techniques, have been discussed below:

- **Gaussian Blur**: After using Gaussian blur, it was observed that the weight threshold should be set between 0.25 and 0.38 and the number of iterations should lie between 30 and 75. As the number of iterations reaches about 80, it starts causing some ringing effect. In this work the best result has been obtained with the number of iterations are 35 and weight threshold is 0.34.
- **Average Blur**: While applying average blur technique the weight threshold should be set between 0.29 and 0.40 and the number of iterations should be between 20 and 70. Because as the number of iterations

reaches near 70, it causes some ringing effect again. In this work the best result has been obtained when the numbers of iterations are 32 and weight threshold is 0.33.

It was observe that Blind Deconvolution gives better performance when it restores the image from Average blur, in terms of temporal efficiency and quality of image.

Conclusion

Restoration technology of image is one of the important technical areas in image processing. In this work, the process of image restoration technique has been discussed. The process is based on Blind Deconvolution approach with partial information available about true image. Advantage of using Blind Deconvolution Algorithm is to deblur the degraded image without prior knowledge of PSF and additive noise. The method differs from most of other existing methods by only imposing weak restrictions on the blurring filter, being able to recover images which have suffered a wide range of degradations. The advantage of the proposed Blind Deconvoultion Algorithm is to deblur the degraded image without prior knowledge of PSF and additive noise. But in other algorithms, to process the image the prior knowledge of blurring parameter is must.

REFERENCES

Antonin Chambolle, Ronald A DeVore, Namyong Lee and Bradley J Lucier (1998). Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing* **7**(3) 319–335, 1998.

Dal Maso G, Jean-Michel Morel and Sergio Solimini (1992). A variation method in image segmentation: existence and approximation results. *Acta Mathematica* **168** 89-151.

David Mumford and Jayant Shah (1989). Optimal approximation by piecewise smooth functions and associated variation problems. *Communications on Pure and Applied Mathematics* **42** 577-685.

Kaleem Siddiqi, Yves B Lauziere, Allen Tannenbaum and Steven W Zuker (1998). Area and length minimizing flows for shape segmentations. *IEEE Transactions on Image Processing* **7** 433-443.

Kundur D and Hatzinakos D (1996). Blind Image Deconvolution. *IEEE Signal Processing Magazine* **13**(3) 43-64.

Rafael C Gonzalez and Richard E Woods (2001). *Digital Image Processing* 1st edition (Addison-Wesley).

Rafael C Gonzalez and Richard E Woods (2009). *Digital Image Processing* 3rd edition (Addison-Wesley).

Ronald A DeVore, Bjorn Jawerth and Bradley J Lucier (1992). Image compression through wavelet transform coding. *IEEE Transactions on Information Theory* **38**(2) 719–746.

Ronald A DeVore, Bjorn Jawerth and Vasil Popov (1992). Compression of wavelet coefficients. *American Journal of Mathematics* **114** 737–785.

Russ JC (1995). *The Image Processing Handbook* 2nd edition (Boca Raton, CRC Press) 674.

Tony F Chan and Luminita A Vese (2001). Active contours without edges. *IEEE Transactions on Image Processing* **10**(2) 266-277.