TJES

ISSN: 1813-162X
Tikrit Journal of Engineering Sciences
available online at: http://www.tj-es.com

Image Deblurring and Effect of Wiener, Regularized, Lucky-Richardson, Blind-Deconvolution, Average and Median Filters on Blurred Image and Noisy Image

Shayma Wail Nourildean*   Mohammed Ismaeel Khalil*

**Al-Dour Technical Institute*

ABSTRACT

In image deblurring, we try to recover the original, sharp image by using a mathematical model of the blurring process. There are several techniques to recover the original image, but they could not recover the image exactly. In this paper, four algorithms (Weiner filter, regularized filter, lucky- Richardson and blind deconvolution), average and median filters were applied on the blurred image to recover the original image. Then noise was added to the original image and the same six algorithms were applied to noisy image. This study was done using MATLAB program, the result shows that the four algorithms (Weiner filter, regularized filter, lucky- Richardson and blind deconvolution) were used for deblurring image with information about distortion operator but median and average filters were failed in deblurring image while they were successful in restoring the noisy image. Wiener and regularized filters were failed to restore the original image from noisy image.

KEYWORDS: Image, Blurred, Noisy, Average, Median

فك طمس او تمويه الصورة وتأثير مرشح الوينر ، المرشح المنظم، خوارزمية ريتشاردسن، خوارزمية بلايند ديكونفوليوشن ، المرشح المعدل والمرشح الوسطي على الصورة المشوهة والصورة الصاخبة

الخلاصة

في فك طمس او تمويه الصورة، نحن نحاول استعادة الصورة الأصلية، الحادة وباستخدام نموذج رياضي لعملية الطمس. هناك أساليب عدة لاستعادة الصورة الأصلية. ولكن لا يمكنهم استعادة الصورة بالضبط. في هذه البحث، الخوارزميات الاربعة (مرشح وينر، المرشح المنظم، خوارزمية ريتشاردسون وخوارزمية blind deconvolution) والمرشح المعدل والوسيط تم تطبيقها على الصورة المطموسة لاستعادة الصورة الأصلية. ثم أضيف الضوضاء إلى الصورة الأصلية، وطبقت نفس الخوارزميات الستة للصورة الصاخبة. أجريت هذه الدراسة باستخدام برنامج MATLAB، النتيجة تظهر ان الخوارزميات الأربعة (مرشح وينر، المرشح المنظم، خوارزمية ريتشاردسون وخوارزمية blind deconvolution) كانت تستخدم لفك طمس الصورة مع معلومات عن مشغل تشويه لكن المرشحات المعدل والوسيط فشلت في اعادة الصورة الاصلية في حين أنها كانت ناجحة في استعادة الصورة الصاخبة. وفشلت مرشح وينر والمرشح المنظم لاستعادة الصورة الاصلية من الصورة الصاخبة.

الكلمات الدالة: صورة، مطموس (مشوه)، صاخب، معدل، وسطي

INTRODUCTION

Interest in digital image processing methods stems from two principal application areas; improvement of pictorial information for human interpretation; and processing of image data for storage. transmission, and representation for autonomous machine perception[1]. This paper introduced the problem of image deblurring and the four algorithms that is used as an attempt to restore the original image. Compared with deblurring using adaptive and median filters which is more effective with noisy image. In this paper, the same four algorithms (Weiner filter, regularized filter, lucky- Richardson and blind deconvolution) and median filter, average filter were applied to noisy image to show the effect of these algorithms on the blurred and noisy image**.**

What Is Digital Image Processing?

The field of digital image processing refers to processing digital images by means of a digital computer. 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 processing on images involves 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 vision[1].

Examples of Fields that Use Digital Image Processing:

Gamma-Ray and X-ray Imaging, Imaging in the Ultraviolet Band, Visible and Infrared Bands, the Microwave Band, the Radio Band [1].

Digital Image Definitions

An image may be defined as a two-dimensional function. f(x, y), where x and y are spatial (plane) coordinates, and the amplitude at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the amplitude values of f are all finite, discrete quantities, the image is called a digital image[1]. The 2D continuous image a(x,y) is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. In fact, in most cases a(x,y) is actually a function of many variables including depth (z), color (λ), and time (t)[2].

Fundamentals Steps of Digital Image Processing

- Image acquisition is the first process that could be as simple as being given an image that is already in digital form. Generally. The image acquisition stage involves preprocessing, such as scaling[1].
- Image enhancement is the process by which we try to improve an image so that it looks subjectively better. We do not really know how the image should look, but we can tell whether more detail can be seen, or whether unwanted flickering has removed, or the contrast is better[3].
- Image restoration is an area that also deals with improving the appearance of an image [2].
- Color image processing is an area that has been gaining in importance due to the significant increase in the use of digital images over the Internet [2].
- Wavelets are the foundation for representing images in various degrees of resolution [2].
- Compression deals with techniques for reducing, the storage required to save an image. or the bandwidth required to transmit it [2].
- Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape [2].
- Image Segmentation and Edge Detection is the process to extract the outlines of different regions in the image; i.e. to divide the image in to regions which are made up of pixels which have something in common. For example, they may have similar brightness, or color, which may indicate that they belong to the same object or facet of an object [3].
- Representation and description almost always follow the output of a segmentation stage, which usually is raw

pixel data. Constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself. Description, also called feature selection, deals with extracting attributes that result in some quantitative information of interest or arc basic for differentiating one class of objects from another [2].

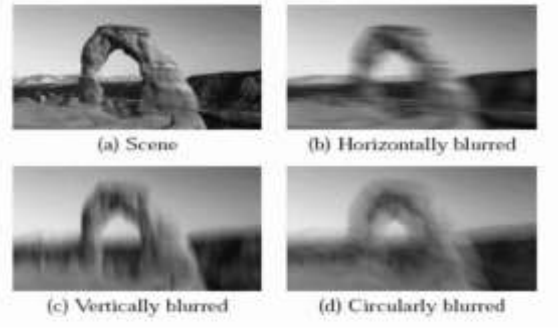
Recognition is the process that assigns a label (e.g., "vehicle") to an object based on its descriptors [2].

IMAGE BLUR

The image formed in the focal plane is blurred by the imaging instrument and the atmosphere. It can be expressed as an integral over the true image, denoted symbolically by $\otimes$:

$$M(\mathbf{x}) = P \otimes I = \int P(\mathbf{x}, \mathbf{y}) I(\mathbf{y})\, d\mathbf{y}.$$

For a 2D image, the integration is over the 2D y-space upon which the image is defined. In general, the imaging problem may be defined in a space with an arbitrary number of dimensions, and the dimensionalities of x and y need not even be the same (e.g., in tomography). Wavelength and/or time might provide additional dimensions. The kernel of the integral, P(x, y), is called the point-spread function. It is the probability that a photon originating at position y in the image plane ends up at position x in the focal plane [4]. When a photograph is taken in low light conditions or of a fast moving object, motion blur can cause significant degradation of the image. This is caused by the movement of the object relative to the sensor in the camera during the time the shutter is open. Both the object moving and camera shake contribute to this blurring. The problem is particularly apparent in low light conditions when the exposure time can often be in the region of several seconds[5]. Motion Blur was shown in Fig.1 [6].



(a) Scene (b) Horizontally blurred

(c) Vertically blurred (d) Circularly blurred

Fig. 1: Motion Blur [6]

The corresponding motion blur is usually modeled as a linear image degradation process [7].

$$I = L \otimes f + n \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where I, L, and n represent the degraded image, unblurred (or latent) image, and the additive noise respectively. $\otimes$ is the convolution operator and f is an unknown linear shift invariant point spread function (PSF). One example of the camera motion blur is shown in Fig.2 (a). The problem of restoring a still image containing a motion blurred object cannot be completely solved by the blind deconvolution techniques because the background may not undergo the same motion. The PSF has a uniform definition only on the moving object. We show one example in Fig. 2 (b) where the motion of an athlete causes the blur while the background is still clear [7]. There are several techniques for either preventing image motion blurring at the time of image capture or Post processing images to remove motion blur. As well as in every day photography, the problem is particularly important to applications such as video surveillance where low quality cameras are used to capture sequences of photographs of moving objects (usually people). Current techniques can be split roughly into the following categories [5]:

1- Hardware in the optical system of the camera to stabilize the image
2- Post processing of the image to remove motion blur by estimating the camera's motion
3- From a single photograph (blind de-convolution)
4- From a sequence of photographs.

5- A hybrid approach that measures the camera's motion during photograph capture.

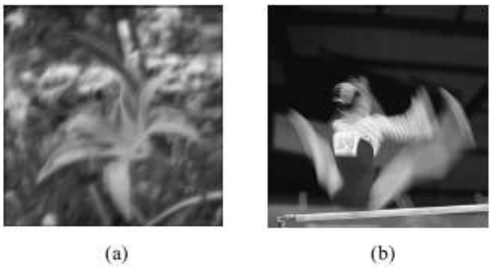


Fig. 2: Two types of motion blur[7].

In image deblurring, we seek to recover the original, sharp image by using a mathematical model of the blurring process. The key issue is that some information on the lost details is indeed present in the blurred image—but this information is “hidden” and can only be recovered if we know the details of the blurring process. Unfortunately there is no hope that we can recover the original image exactly! This is due to various unavoidable errors in the recorded image. The most important errors are fluctuations in the recording process and approximation errors when representing the image with a limited number of digits. The influence of this noise puts a limit on the size of the details that we can hope to recover in the reconstructed image, and the limit depends on both the noise and the blurring process. MATLAB is an excellent environment in which to develop and experiment with filtering methods for image deblurring. The basic MATLAB package contains many functions and tools for this purpose, but in some cases it is more convenient to use routines that are only available from the Signal Processing Toolbox (SPT) and the Image Processing Toolbox (IPT). We will therefore use these toolboxes when convenient [8].

Deblurring Model

A blurred or degraded image can be approximately described by the following equation [9]:

$$g = Hf + n \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Where:

g is the blurred image

H is the distortion operator, also called the point spread function (PSF).

f The original true image

n Additive noise, introduced during image acquisition and corrupt image.

In the spatial domain, the PSF describes the degree to which an optical system blurs (spreads) a point of light. The PSF is the inverse Fourier transform of the optical transfer function (OTF). In the frequency domain, the OTF describes the response of a linear, position-invariant system to an impulse. The OTF is the Fourier transform of the point spread function (PSF). The distortion operator, when convolved with the image, creates the distortion. Distortion caused by a point spread function is just one type of distortion [9].

Importance of the PSF

Based on this model, the fundamental task of deblurring is to deconvolve the blurred image with the PSF that exactly describes the distortion. Deconvolution is the process of reversing the effect of convolution. To illustrate, this example takes a clear image and deliberately blurs it by convolving it with a PSF. The example uses the fspecial function to create a PSF that simulates a motion blur, specifying the length of the blur in pixels, (LEN=31), and the angle of the blur in degrees (THETA=11). Once the PSF is created, the example uses the imfilter function to convolve the PSF with the original image, I, to create the blurred image, Blurred[9]. The following matlab program is an example of blurring process[9]:

```
I =imread('liftingbody.png');   % read the
image

I=I(60+[1:256],222+[1:256],:); % crop the
image

figure; imshow(I); title('Original Image');  %
show the original image

LEN = 31;  % the length of image in pixles

THETA = 11;  % angle of blur in degrees

PSF = fspecial('motion',LEN,THETA);    %
create PSF(distortion operator)

Blurred = imfilter(I,PSF,'circular','conv'); %
blurring the image

figure; imshow(Blurred); title('Blurred
Image');  % show the blurred image
```

The results of the original and blurred images were shown in Fig. 3

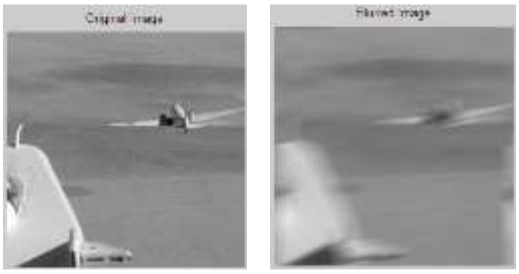


Fig.3 Original and Blurred Image [9]

Deblurring Algorithms:

There are four algorithms used for debluring the image as follows:

Deblurring with the Wiener Filter

Wiener theory, formulated by Norbert Wiener, forms the foundation of data-dependent linear least square error filters. Wiener filters play a central role in a wide range of applications such as linear prediction, echo cancellation, signal restoration, channel equalization and system identification[10]. Use the deconvwnr function to deblur an image using the Wiener filter. Wiener deconvolution can be used effectively when the frequency characteristics of the image and additive noise are known, to at least some degree. In the absence of noise, the Wiener filter reduces to the ideal inverse filter[9].

Deblurring with a Regularized Filter:

A regularized filter can be used effectively when constraints are applied on the recovered image (e.g., smoothness) and limited information is known about the additive noise. . The blurred and noisy image is restored by a constrained least square restoration algorithm that uses a regularized filter[9].

Deblurring with the Lucy-Richardson Algorithm:

The Lucy-Richardson algorithm can be used effectively when the point-spread function PSF (blurring operator) is known, but little or no information is available for the noise. The blurred and noisy image is restored by the iterative, accelerated, damped Lucy-Richardson algorithm. The additional optical system (e.g. camera) characteristics can be used as input parameters to improve the quality of the image restoration. Lucy-Richardson algorithm. The algorithm maximizes the likelihood that resulting resulting image, when convolved with the PSF, is an instance of the blurred image, assuming Poisson noise statistics. This function can be effective when you know the PSF but know little about the additive noise in the image [9].

Deblurring with the Blind Deconvolution Algorithm

deconvolution is the process of removing the blur function from an image. Blind deconvolution methods are used in cases where the blur function cannot be measured or accurately modeled. In some cases, the application of a sharpening filter will suffice. The objective of blind deconvolution is to find the best estimate of original image from blurred image when the PSF and noise function are unknown [11].

The algorithm maximizes the likelihood that the resulting image, when convolved with the resulting PSF, is an instance of the blurred image, assuming Poisson noise statistics. The blind deconvolution algorithm can be used effectively when no information about the distortion (blurring and noise) is known. The deconvblind function restores the image and the PSF simultaneously, using an iterative process similar to the accelerated, damped Lucy-Richardson algorithm [9]. The deconvblind function, just like the deconvlucy function, implements several adaptations to the original Lucy-Richardson maximum likelihood algorithm that address complex image restoration tasks. Using these adaptations, you can [9].

- Reduce the effect of noise on the restoration.
- Account for nonuniform image quality (e.g., bad pixels). Handle camera read-out noise.

SIMULATION RESULTS

Matlab program[9] for the four deblurring algorithms (Weiner filter, regularized filter, lucky- Richardson and blind deconvolution) had been modified to be applied on blurred and noisy images. In addition, the matlab program [9] for average and median filters had been modified to be applied on blurred and noisy images. In this paper, these programs were taken in four simulation procedures to study the effect of these six algorithms on blurred and noisy image to show the difference between them as follows:

Procedure 1

In this procedure, the image was blurred by convolving the image with the distortion operator (PSF). The four algorithms (Weiner filter, regularized filter, lucky- Richardson and blind deconvolution) were applied to deblur the image to show the effect of these algorithms on the blurred image as an attempt to restore the original image.

Wiener Filter

The image was blurred with the same PSF. The wiener filter was applied on the blurred image. The matlab code is:

```
I = imread('testpat1.png'); % read the image
figure;imshow(I);title('Original Image'); %show the original image
LEN = 31; % the length of image in pixles
THETA = 11; % angle of blur in degrees
PSF = fspecial('motion',LEN, THETA); %create PSF (distorsion operator)
Blurred = imfilter(I,PSF,'circular','conv');%blurring the image
figure; imshow(Blurred);title('Blurred Image');%show the blurred image
wnr1 = deconvwnr(Blurred,PSF); % use deconvwnr function to deblur the image
figure;imshow(wnr1); % show the deblurred image
title ('Restored, True PSF');
```

the results of original image, blurred image, and restored image were shown in Fig. 4

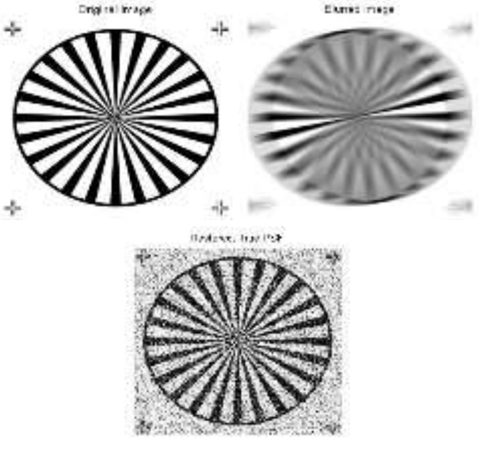


Fig. 4 Original, blurred and restored image using Wiener filter

Regularized Filter

The image was blurred with the same PSF. The regularized filter was applied on the blurred image. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image'); %show the original image
LEN = 31; % the length of image in pixles
THETA = 11; % angle of blur in degrees
PSF = fspecial('motion',LEN, THETA); %create PSF (distorsion operator)
Blurred = imfilter(I,PSF,'circular','conv'); %blurring the image
figure; imshow(Blurred);title('Blurred Image');%show the blurred image
[reg1 LAGRA] = deconvreg(Blurred,PSF);%deconvreg function to deblur the image
figure,imshow(reg1),title('Restored Image'); % show the restored image
```

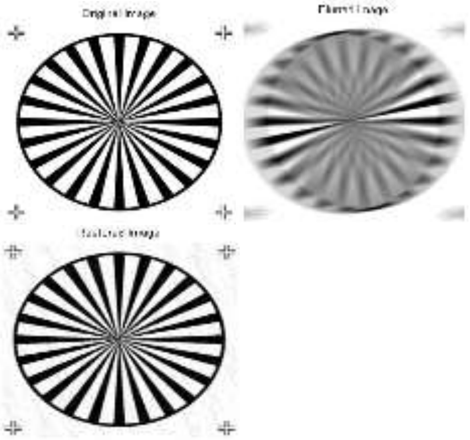


Fig. 5 Original, blurred and restored image using Regularized filter

Lucy-Richardson Algorithm

The deconvlucy function implements several adaptations to the original Lucy-Richardson maximum likelihood algorithm that address complex image restoration tasks [9]. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image'); %show the original image
LEN = 31; % the length of image in pixles
THETA = 11;% angle of blur in degrees
PSF = fspecial('motion',LEN, THETA); %create PSF (distorsion operator)
Blurred = imfilter(I,PSF,'circular','conv'); %blurring the image
figure; imshow(Blurred);title('Blurred Image');%show the blurred image
```

```
luc1 = deconvlucy(Blurred,PSF); % use deconvlucy to deblur image
figure; imshow(luc1); title('Restored Image'); % show the image
```

The results of original image, blurred image, and restored image were shown in Fig. 6.

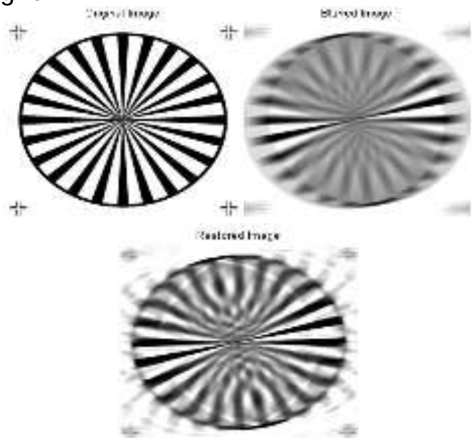


Fig. 6 Original, blurred and restored image using Lucy-Richardson Algorithm

Blind Deconvolution Algorithm

Use the deconvblind function to deblur an image using the blind deconvolution algorithm. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image'); %show the original image
LEN = 31; % the length of image in pixles
THETA = 11; % angle of blur in degrees
PSF = fspecial('motion',LEN, THETA); %create PSF (distorsion operator)
Blurred = imfilter(I,PSF,'circular','conv'); %blurring the image
figure; imshow(Blurred);title('Blurred Image'); %show the blurred image
INITPSF = ones(size(PSF));
[J P]= deconvblind(Blurred,INITPSF,30); % use deconvlucy to deblur image
figure; imshow(J); title('Restored Image'); % show the image
```

The results of original image, blurred image, and restored image were shown in Fig. 7

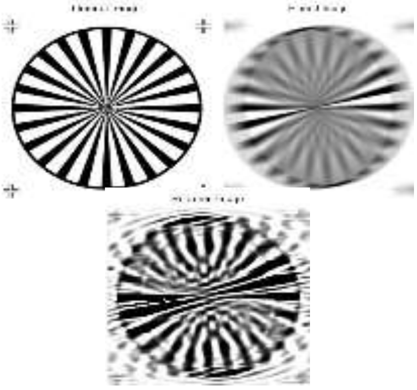


Fig. 7 Original, blurred and restored image using Blind Deconvolution algorithm

The median and average filters would be applied to the blurred image as shown in the following results:

Deblurring the image with average filter

The same image was deblurred with the same PSF and the average filter was applied. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image'); %show the original image
LEN = 31; % the length of image in pixles
THETA = 11; % angle of blur in degrees
PSF = fspecial('motion',LEN,THETA); %create PSF (distorsion operator)
Blurred = imfilter(I,PSF,'circular','conv'); %blurring the image
figure; imshow(Blurred);title('Blurred Image'); %show the blurred image
L = filter2(fspecial('average',3),Blurred)/255; % apply average filter
 figure, imshow(L); title('Average Filter to restore blurred image'); % show the image
```

The results of original image, blurred image, and restored image were shown in Fig. 8.

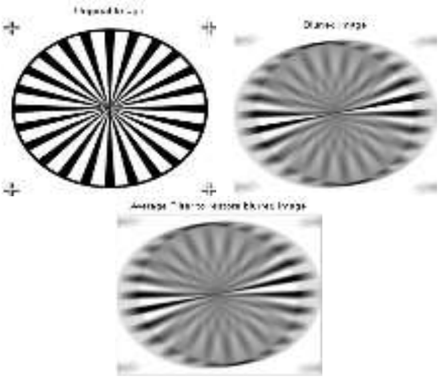


Fig. 8 Original, blurred and restored image using Average filter

Deblurring the image with median filter

The same image was deblurred with the same PSF and the average filter was applied. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image'); %show the original image
LEN = 31; % the length of image in pixles
THETA = 11; % angle of blur in degrees
PSF = fspecial('motion',LEN,THETA); %create PSF (distorsion operator)
Blurred = imfilter(I,PSF,'circular','conv'); %blurring the image
figure; imshow(Blurred);title('Blurred Image'); %show the blurred image
M = medfilt2(Blurred,[3 3]); % apply median filter
 figure, imshow(M); title('Median Filter to restore blurred image'); % show the image
```

The results of original image, blurred image, and restored image were shown in Fig. 9.

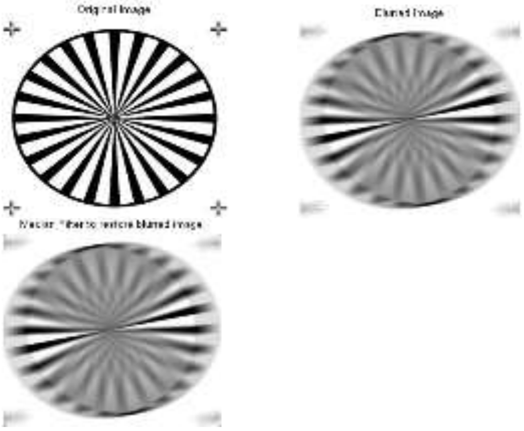


Fig. 9 Original, blurred and restored image using Median filter

Procedure 2

The Gaussian noise with variance V = 0.2 was added to the original image to create noisy image. The same algorithms were applied to the noisy image as an attempt to restore the noisy image.

Wiener filter

The wiener filter was applied to noisy image which cause the distortion image. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image'); %show the original image
V = .02;  % Variance of Gaussian noise
PSF = fspecial('gaussian',11,5); %create PSF (distorsion operator)
Noisy = imnoise(I,'gaussian',0,V); % create noisy image
figure;imshow(Noisy);title('Noisy Image'); % show the noisy image
wnr1 = deconvwnr(Noisy,PSF);   % use deconvwnr function to deblur the image
figure;imshow(wnr1); title('Restored Image');% show the restored image
```

The results of original image, noisy image, and restored image were shown in Fig. 10.

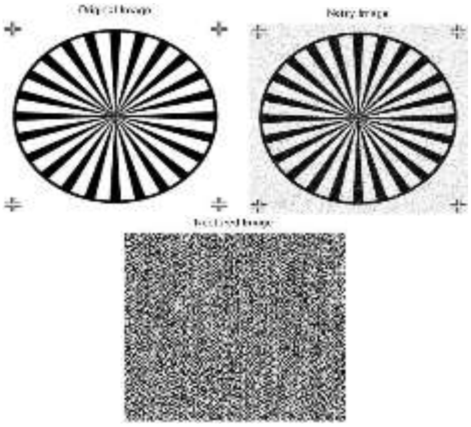


Fig. 10 Original, Noisy and restored image using Wiener filter

Regularized filter

The wiener filter was applied to noisy image. The matlab code is:

```
I = imread('testpat1.png'); %read the image
figure;imshow(I);title('Original Image');%show the original image
V = .02; %Variance of Gaussian noise
PSF =fspecial('gaussian',11,5); %create PSF(distortion operator
Noisy = imnoise(I,'gaussian',0,V); %create noisy image
figure;imshow(Noisy); title('Noisy Image');%show noisy image
NP = V*prod(size(I));
[reg1 LAGRA] = deconvreg(Noisy,PSF,NP); % use deconvreg function to deblur the image
figure,imshow(reg1),title('Restored Image'); % show the restored image
```

The results of original image, noisy image, and restored image were shown in Fig. 11.

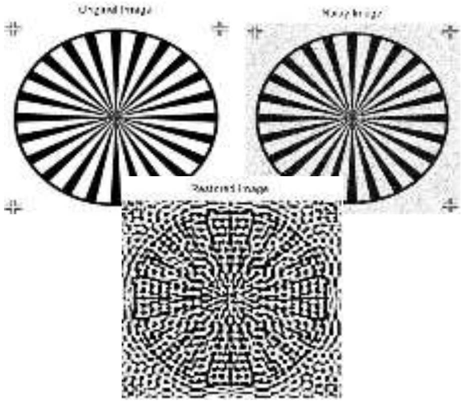


Fig. 11 Original, Noisy and restored image using Regularized filter

Lucky- Richardson

The Lucky- Richardson was applied to noisy image. The matlab code is:

```
I = imread('testpat1.png');                %read the image
figure;imshow(I);title('Original Image');%show the original image
V = .02;% Variance of Gaussian noise
PSF = fspecial('gaussian',11,5); %create PSF(distortion operator)
Noisy = imnoise(I,'gaussian',0,V);          % create noisy image
figure;imshow(Noisy); title('Noisy Image');% show noisy image
luc1 = deconvlucy(Noisy,PSF,5);%use deconvlucky function to restore the image
figure,imshow(luc1),title('Restored Image');% show the restored image
```

The results of original image, noisy, and restored image were shown in Fig. 12

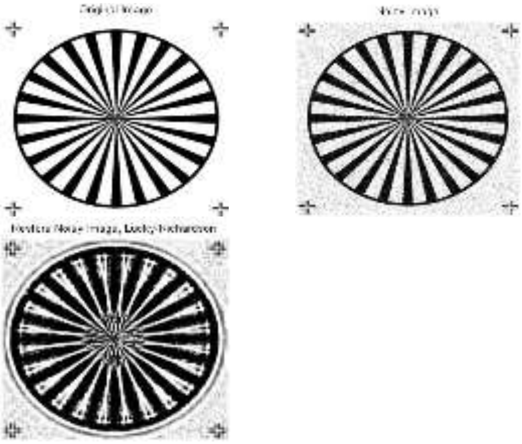


Fig. 12 Original, Noisy and restored image using Lucy-Richardson Algorithm

Blind-deconvolution

The Blind-deconvolution was applied to noisy image.

The matlab code is:

```
I = imread('testpat1.png');  %read the image
figure;imshow(I);title('Original Image'); %show the original image
V = .02;  % Variance of Gaussian noise
PSF = fspecial('gaussian',11,5);   %create PSF (distorsion operator)
Noisy = imnoise(I,'gaussian',0,V); % create noisy image
figure;imshow(Noisy); title('Noisy Image'); % show noisy image
[J P]= deconvblind(Noisy,PSF); % use deconvblind function to deblur the image
figure,imshow(J),title('Restored Image');  % show the restored image
```

The results of original image, blurred image, and restored image were shown in Fig. 13

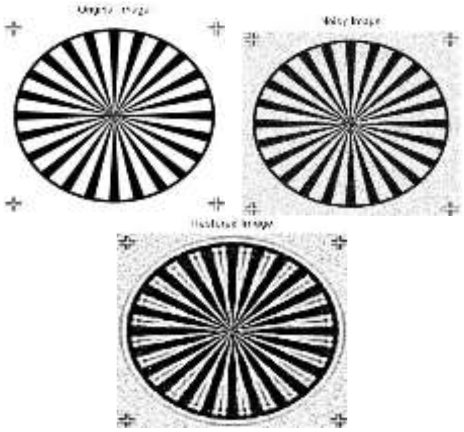


Fig. 13 Original, Noisy and restored image using Blind Deconvolution algorithm

Average Filter

The average filter was applied to noisy image. The matlab code is:

```
I = imread('testpat1.png');                %read the image
figure;imshow(I);title('Original Image'); %show the original image
V = .02;  % Variance of Gaussian noise
PSF = fspecial('gaussian',11,5); %create PSF (distortion operator)
Noisy = imnoise(I,'gaussian',0,V);          % create noisy image
figure;imshow(Noisy); title('Noisy Image');% show noisy image
L = filter2(fspecial('average',3),Noisy)/255;% apply average filter
figure, imshow(L); title('Average Filter to restore image');% show the image
```

The results of original image, blurred image, and restored image were shown in Fig. 14

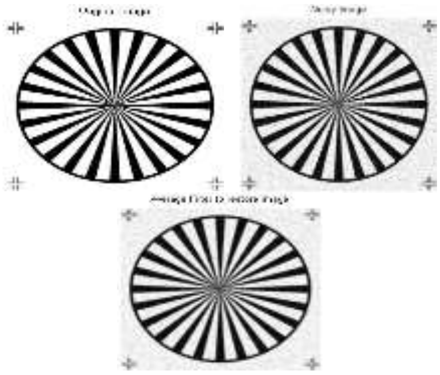


Fig. 14 Original, Noisy and restored image using Average filter

<u>Median Filter:</u>

The median filter was applied to noisy image. The matlab code is:

```
I = imread('testpat1.png');                %read the image
figure;imshow(I);title('Original Image'); %show the original image
V = .02; % Variance of Gaussian noise
PSF = fspecial('gaussian',11,5); %create PSF (distortion operator)
Noisy = imnoise(I,'gaussian',0,V);  % create noisy image
figure;imshow(Noisy); title('Noisy Image');% show noisy image
M = medfilt2(Noisy,[3 3]); % apply median filter
figure, imshow(L); title('Median Filter to restore image'); % show the image
```

the results of original image, blurred image, and restored image were shown in Fig. 15

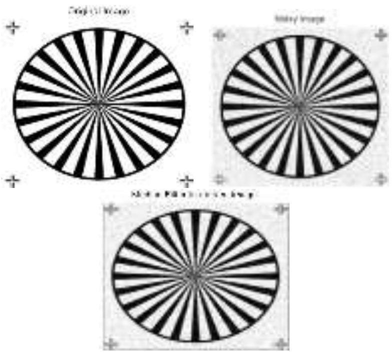


Fig. 15 Original, Noisy and restored image using Median filter

Procedure 3

The matlab programs for the Wiener Filter, Regularized Filter, Lucky-Richardsom, Blind-deconvolution, average filter and median filters were assembled in the same program to restore the blurring image for the purpose of comparison between the effect of them on the blurred image as in the following matlab code:

```
I = imread('testpat1.png');
figure;imshow(I);title('Original Image');
LEN = 31;
THETA = 11;
PSF = fspecial('motion',LEN,THETA);
Blurred = imfilter(I,PSF,'circular','conv');
figure;          imshow(Blurred);title('Blurred Image');
wnr1 = deconvwnr(Blurred,PSF);
figure;imshow(wnr1);
title('Restored, Weiner Filter');
[reg1 LAGRA] = deconvreg(Blurred,PSF);
figure,imshow(reg1),title('Restored    Image, Regularized Filter');
luc1 = deconvlucy(Blurred,PSF);
title('Restored Image, Lucky-Richardson');
INITPSF = ones(size(PSF));
[J P]= deconvblind(Blurred,INITPSF,30);
figure; imshow(J); title('Restored Image, Blind Deconvolution');
L   =   filter2(fspecial('average',3),Blurred) /255;
figure, imshow(L); title('Average Filter to restore blurred image');
M = medfilt2(Blurred,[3 3]);
 figure, imshow(M); title('Median Filter to restore blurred image');
```

The image results were shown in Fig. 16

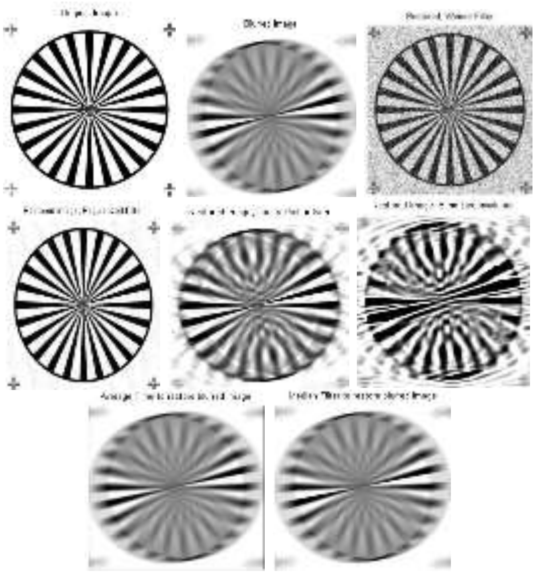


Fig.16 Original, Blurred, Restored image with Wiener filter, Regularized filter, Lucky-Richardson, Blind deconvolution, Average filter, Median filter

Procedure 4

The matlab programs for the Wiener Filter, Regularized Filter, Lucky-Richardsom, Blind-deconvolution, average filter and median filters were assembled in the same program to restore the original image from the noisy image for the purpose of comparison between the effect of them on noisy image as in the following matlab code:

```
I = imread('testpat1.png');
figure;imshow(I);title('Original Image');
V = .02;
PSF = fspecial('gaussian',11,5);
Noisy = imnoise(I,'gaussian',0,V);
figure; imshow(Noisy);title('Noisy Image');
wnr1 = deconvwnr(Noisy,PSF);
figure;imshow(wnr1);
title('Restore Noisy Image, Weiner Filter');
[reg1 LAGRA] = deconvreg(Noisy,PSF);
figure,imshow(reg1),title('Restore Noisy Image, Regularized Filter');
luc1 = deconvlucy(Noisy,PSF);
figure; imshow(luc1);
title('Restore Noisy Image, Lucky-Richardson');
[J P]= deconvblind(Noisy,PSF);
figure; imshow(J); title('Restore Noisy Image, Blind Deconvolution');
L = filter2(fspecial('average',3),Noisy)/255;
figure, imshow(L); title('Average Filter to restore Noisy image');
M = medfilt2(Noisy,[3 3]);
figure, imshow(M); title('Median Filter to restore Noisy image');
```

The image results were shown in Fig. 17

DISCUSSION AND CONCLUSION

1- The Wiener Filter algorithm, Regularized filter, the Blind Deconvolution Algorithm and the Lucy-Richardson Algorithm were used for image deblurring as an attempt to restore the original image.

2- The information about the blurring process was important such as the distortion operator (PSF) in order to restore the original image from the blurred and noisy image.

3- From procedure 1, deblurring the image with wiener filter do a good job to restore the image if the PSF was exactly known.

4- From procedure 1, deblurring the image with regularized filter was very effective to restore the original image from the blurred image.

5- From procedure 1, deblurring with the Lucy-Richardson Algorithm was very effective to restore the image when there is little information about the PSF and the blind deconvolution works better when there is no information about the PSF.

6- From procedure 1, average and median filters were failed to restore the original image from the blurred image.

7- Number of iterations could be done to the algorithms (Weiner filter, regularized filter, lucky- Richardson and blind deconvolution) to give better results.

8- From procedure 2, when the noise was added to the image, the wiener and regularized filters were unsuccessful and they cause distortion of the image while they attempt to restore the image. As a result, they cannot remove the noise to restore the original image.

9- From procedure 2, Lucy-Richardson and blind-deconvolution work better to restore the original image from the noisy image when there is information about PSF (distortion operator). They restore the negative of the original image.

10- From procedure 2, average and median filters were successful to restore the original image from the noisy image.

11- From Procedure 3, The comparison between the six algorithms on the same blurred image with the same distortion operator (PSF) was done and showed that the regularized filter was very effective to restore the blurring image. median and average filters have no effect in deblurring the image.

12- From Procedure 4, The comparison between the six algorithms on the same noisy image with the same distortion operator (PSF) was done and showed that the average and median filters work better to restore the original image from the noisy image and median filter was very effective to restore the original image from the noisy image.

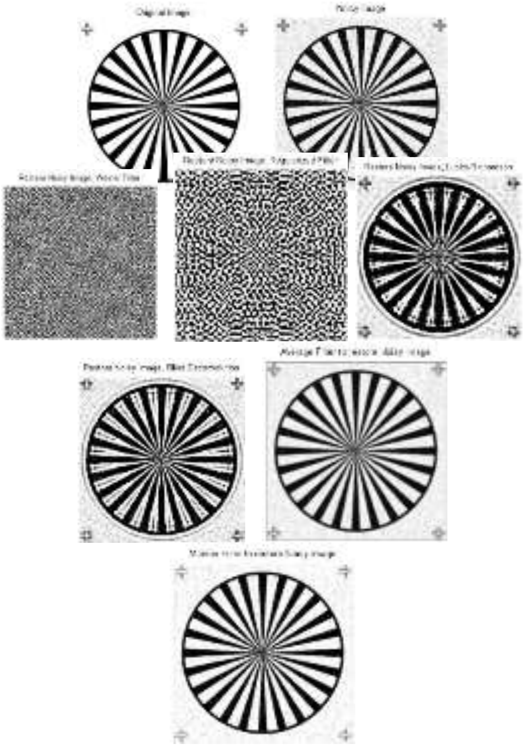


Fig.17 Original, Noisy, Restored image with Wiener filter, Regularized filter, Lucky-Richardson, Blind deconvolution, Average filter, Median filter

REFERENCES

1- Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Prentice Hall, pp 1-28, 2002.

2- Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet, “Fundamentals of Image Processing”, The Netherlands at the Delft University of Technology ISBN 90–75691–01–7, pp 2, 1998.

3- Maria Petrou, “Image Processing, the fundamentals”, John Wiley & Sons Ltd, ISBN: 0 – 471 – 99883 – 4, pp 125, 265, 1999.

4- R.C. Puetter,1,4 T.R. Gosnell,2,4 and Amos Yahil, “DIGITAL IMAGE RECONSTRUCTION: Deblurring and Denoising”, Annu. Rev. Astron. Astrophys.. 43:139–94, University of Califorenia, pp 139 – 194, 2005.

5- K.Srinivasa Reddy,2 N.Udaya Kumar, 3G.Appala Naidu,4 k.Geetha Reddy, “ Blur Image Estimation in the Frequency Domain by using Image Quality Index”, nternational Journal of Engineering and Technology ISSN: 2049-3444,Volume 2 No. 1, January, pp 28-35, 2012.

6- M. BenEzra and S. K. Nayar. Motion Deblurring Using Hybrid Imaging. Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2003.

7- Jiaya Jia, “Single Image Motion Deblurring Using Transparency”, 1-4244-1180-7/07/ ©2007 IEEE, Chinese University of Hong Kong.

8- Christian Hansen, James G. Nagy, and Dianne P. O’Leary , “Deblurring Images: Matrices, Spectra, and Filtering”, Society for Industrial and Applied Mathematics, pp 1-2, 2012.

9- Matlab program, http://www.mathworks.com/

10- Saeed V. Vaseghi, “Advanced Digital Signal Processing and Noise Reduction, Second Edition.”, John Wiley & Sons Ltd, pp 178, 2000.

11- William E. Vanderlinde, James N. Caron, “Blind Deconvolution of SEM Images”, Proceedings of the 33rd International Symposium for Testing and Failure Analysis, November 4-8, pp 97-102, 2007.