Estimating Blur Parameters to Reconstruct the Motion Blurred Images

Nidhal K. El Abbadi[1](✉), Ali Hussein Abdulkhaleq[2], and Safaa Alwan Al Hassani[3]

[1] Computer Techniques Engineering Department, College of Engineering & Technology, Al-Mustaqbal University, Babylon, Iraq
nidhal.abass@fulbrightmail.org
[2] Department of Information Technology, Technical College of Management, Al-Furat Al-Awsat Technical University, Najaf, Iraq
[3] Computer Science Department, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq

Abstract. One of the important topics of image restoration is the restoration of a motion-blurred image. The restored blurred images are based on estimating the two parameters of a point spread function (PSF) angle and length of the blur kernel. In this article, we suggested checking the quality of the input images before going further in processing when the image is not blurred, estimating the PSF parameters, and reconstructing the image based on the periodicity of motion-blurred in the frequency domain. Fourier transform is used to detect the blur in the input images, and the Hough transform is used to estimate the blurred image angle. Also, the cepstrum domain is used for blur length estimation by finding the lowest peak. After blur parameters are estimated, we determine a point spread function (PSF) to deblur the image using a wiener filter of non-blind deconvolution. The visual, as well as the peak signal-to-noise ratio (PSNR) of restored images, are compared with competent recent schemes. The experimental results of the suggested algorithm show good accuracy, robustness, and time efficiency for blur detection, estimation of the PSF parameters, and reconstruction of the image when compared with other works and methods.

Keywords: blur angle · blur length · blur image · image processing · point spread function

1 Introduction

Modern image processing sciences, including photography, astronomical images, medical images, and microscopy images, have evolved well over recent years and many advanced techniques have emerged. These advances have allowed images to be acquired at higher speeds and higher resolution. High-resolution techniques can lead to degraded acquired image quality, which is an example of a blur. The effect of image de-blurring is effectively a clear image when eliminating distortion and blur [1].

A. M. Al-Bakry et al. (Eds.): NTICT 2022, CCIS 1764, pp. 3–24, 2023.
https://doi.org/10.1007/978-3-031-35442-7_1

Image blur is the primary cause of image loss, and the de-blurring image is a common research concern in the field of image processing [2]. Different factors can cause image degradation, such as out of focus, camera movement or object movement, and atmospheric turbulence. Any motion at the time of image capturing for the camera or objects can cause a motion blur, this is because the sensors can observe the same point on the scene [3]. Mathematically, the motion-blurred image can be modeled as a convolution of the latent image with the point spread function (PSF), also called the blur kernel [4]. Blur image restoring, is a process in which latent sharp images are restored from the observed image using part or no information about the blurring process. There are two main methods to restore the images from degraded images, non-blind and blind images. In most cases a PSF is assumed to be known before recovering the true image, this is called the non-blind restoration or classical restoration. While the second method is based on the estimation of PSF without any prior information about the blurring process [2]. Generally, the restoration of motion-blurred images aims to eliminate the effect of PSF. There are two parameters used to determine PSF: the motion length and direction, unfortunately, their values are often unknown because there is no information about motion when the image is captured [3, 4].

De-blurring is a technique to remove artifacts from images that blur. Image deblurring can broadly divide into two classes, specifically blind and non-blind deconvolution. Some of the well-known non-blind de-blurring methods are the Lucy Richardson algorithm, Wiener filter, and Regularized filter. The accuracy of restoration depends on the estimation of blur length and blur angle that constructs the PSF itself. For instance, after the PSF reconstruction from these two parameters, a Wiener filter can restore a degraded image in the current method.

The main contribution of this paper is the ability to check the images whether it is blurred or not before any deblurring process. Also, the high accuracy of estimating the motion blur angle and length is considered a contribution that enhances the deblurring process.

The rest of the paper includes related works in Sect. 2, where there are several previous and related papers discussed. A brief idea about the motion blur, Hough transform, and cepstrum domain analysis is presented in Sects. 3, 4 and 5 respectively. The methodology of the proposed method is introduced in detail in Sect. 6, while the results of the proposed method are displayed in Sect. 7. Finally, introduce the conclusions in Sect. 8.

2 Related Works

This section is focused on the reviews of previous methods that have worked in this field of study, emphasizing the studies with the most relevance to the method suggested in this work.

(Li et al. 2021) Introduced a method to recover the motion blurred image captured by mobile device. They suggested using a generative adversarial network based on the mobile net network. The backbone of the network is already trained (where only the generator is trained), so the entire network is trained by the authors [5].

(Zhang and Zheng, 2020) incorporated the Fourier image spectrum with the identification of edges dependent on phase consistency. The blurring angle is defined by measuring the direction of the central bright stripe's edge. Bilinear interpolation is then

used for producing the sub-pixel image of the spectrum, measuring the distance between the dark bands, and estimating the blurring length [6].

(Y. Elmi et al., 2020) presented an approach for estimating parametric PSF. This approach can be used to estimate the linear motion blur vector parameters. This method is used to determine the angle and length of the motion blur vector in a single image to produce PSF for de-blurring. This blind approach takes small vectors in all trends and deburrs with the PSF of those candidate vectors. A no-reference quality measurement metric assesses the goodness of the deblurred image. The no-reference image quality metric assesses measuring the edge degradation in the sharpness, and the quantity of artifact produced from saturated pixels in the deblurred image. In the next iteration, the unacceptable blurring caused by motion vectors is eliminated. The approximation has been enhanced by extending the length of the rest vectors, and the same procedure is continued recursively. The method continues until only one vector remains as the estimate for movement vector blur. In this method there is no additional hardware is needed, fully automatic, and doesn't require operator intervention, use the blurred image for estimating PSF only, improved movement accuracy blurred vector parameters, and less computational time [7].

(Murthy et al., 2018) introduced a technique used to estimate motion blur parameters by formulating the trigonometric relationship between the movement of blurred spectral lines of the image and the blurred parameters. The length of motion blur is calculated by rotating the Fourier spectrum to the estimated motion angle. This allows forehand angle estimation to be performed. By exploring the trigonometric relation between spectral lines, the suggested approach estimates both length and angle simultaneously, thus eliminating spectrum rotation for length estimation [8].

(Albluwi et al., 2018) proposed a method based on a convolution neural network (CNN) for deblurring images and increasing image resolution of blurred low-resolution images. The super-resolution CNN architecture is composed of four layers, two for input and output and two hidden layers, the first hidden layer focus on extracting the overlapping patch. Each patch represents a high-dimensional vector. While the second hidden layer maps the high dimension vector to another high-dimensional vector to represent a high-resolution patch [9].

(Kumar, 2017) suggested an efficient method of estimating parameters of the point spread function (PSF) based on the Histogram of Oriented Gradients (HoG) and the statistical characteristics of an image. In specific, HoG and statistical characteristics help to estimate the motion-based PSF angle and velocity parameters. Once all the parameters have been estimated, the blurred image reconstruction is carried out using a newly introduced non-blind process in the moment domain. The benefit of using the moment domain is quick convergence, and the variations in the PSF parameters are robust [10].

3 Motion Blur Model

The 2D linear shift-invariant model that case the blur and the degraded image is represented by Eq. 1.

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y) \tag{1}$$

where:

f(x, y): is the original image.
g (x, y): is the blurred image.
h: is PSF.
η (x, y): is Gaussian noise.

The angle and length of linear blur motion are related directly to the camera properties during the time of the image captured. After estimating the two parameters of the blur kernel's length and angle, PSF could be reconstructed, and a blurred image restoration could be done [4].

4 Hough Transform

The Hough transform is a technique used to find shapes in a digital image. The classical Hough transform is most commonly used to detect regular curves such as lines, circles, ellipses, etc. In this paper, we use a polar mathematical model (Eq. 2) to detect lines:

$$\rho = x * \cos(\theta) + y * \sin(\theta) \tag{2}$$

where ρ (rho) is the distance of the perpendicular vector from the origin to the line, ϴ (theta) is the angle between the x-axis and the ρ vector. The values ρ and θ are limited to θ ∈ [0, 180] in degrees, and ρ ∈ [−D, D] where D is the maximum distance between opposite corners in an image. A line in image space can be represented as a single point in the parameter space (also called Hough space) with the parameters θ and ρ. Each point in the image space can produce a sinusoid curve in the Hough space. A line in image space consists of many pixels or points, these points can define curves in the Hough space, and all the lines (belonging to points of a single line in image space) in the Hough space pass through a single point. When the Hough transform is applied to the image for the entire pixels, then it is easy to detect the lines that have the high vote. The result of the Hough transform is stored in a matrix that is often called an accumulator table. Where one dimension of this matrix is the theta values (angles), and the other dimension is the rho values (distances) [11].

5 Cepstrum Domain Analysis

Cepstrum transform is used to separate the blur components from the image components. Without the additive noise, image g (x, y) is defined in the Cepstrum transform as follows:

$$Cp = F^{-1} log|F\{G(x,\}|) \tag{3}$$

$$= F^{-1}\{\log[H(u,v)].log[F(u,v)]\}$$

$$= F^{-1}\{\log[H(u,v)] + log[F(u,v)]\}$$

$$= C_h(p,q) + C_f(p,q)$$

where F {.} denotes Fourier transform, G (u, v), H (u, v), and F (u, v) are Fourier transforms of the blurred image, PSF, and the original image respectively. As a blurred image, it is better to separate motion blur from the cepstrum domain than the Frequency domain. In the frequency domain, it is possible to represent *sinc* function. So, uniform motion blur has periodic patterns by zero-crossing of *sinc* function, this is displayed in the cepstrum domain as negative peaks as shown in Fig. 1, where the original image blurred with uniform motion blur by an angle equal to 45° and length equal 15. The *sinc* function in the frequency domain is shown in Fig. 1 (c), and the negative peaks of the blurred image's cepstrum are shown in Fig. 1 (d) [12].

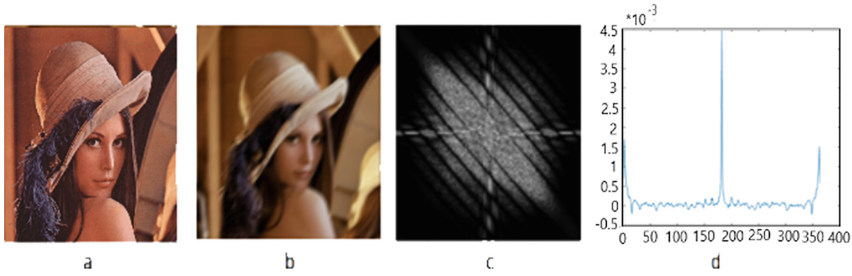

Fig. 1. (a) Original image (b) Blurred image (c) Fourier transform (d) Negative peaks.

6 Methodology

Blurring the image with a linear motion makes the edges in the image expand in the direction specified by the PSF. This research aims to determine the length and direction of the edge extension which represents the PSF parameters. The proposed system consists of four main stages: blur detection, blur angle estimation, blur length estimation, and construction of the PSF algorithm. The construct of the PSF algorithm depends on the blur angle algorithm and the blur length algorithm. Figure 2 shows a flowchart of the proposed system.

6.1 Check the Image Quality

The first step in the proposed algorithm is to check the quality of the input image, when the input image is blurred then continue with other steps; otherwise, when the image has no blur (clear) then no need to remove the blur. Algorithm 1 summarized the main steps for checking the image quality.

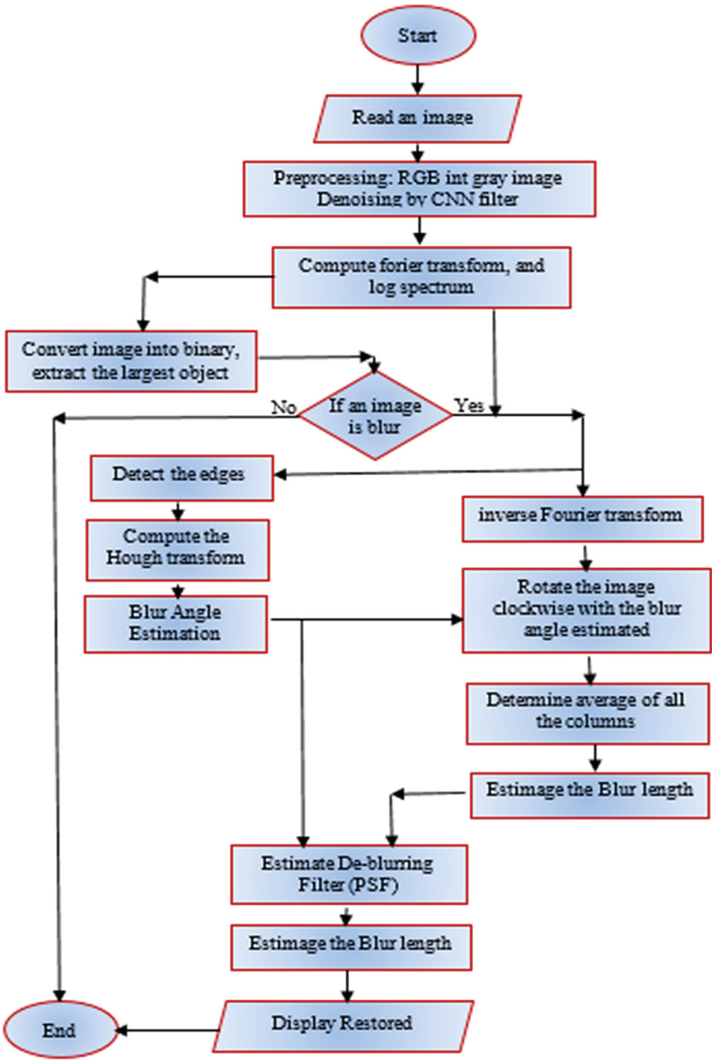

Fig. 2. General flowchart of the proposed system

Algorithm 1: Checking Image Quality

Input: Image
Output: Decision "Blur or Clear image"

Begin
Step 1: Input color image.
Step 2: Convert RGB image to grayscale image.
Step 3: Compute the Fourier transform F (u, v).
Step 4: Compute the log spectrum of F (u, v).
Step 5: Convert the image to a binary image.
Step 6: Extract the largest object (connected component)
Step 7: Plot an ellipse around the selected object.
Step 8: Determine whether the input image is blurry or not by calculating The length of the long diagonal of the ellipse and comparing it with the threshold values.
End.

If the input image is colored, then it is converted into a grayscale image. The suggested algorithm works efficiently with the grayscale images. The gray image will be transformed by using discrete Fourier transformation. In this step, horizontal lines of the spectrum (in dark, light, and white strips) are obtained in case of a blurry image, but these lines are not clear. So, a logarithmic transformation is applied to the spectrum to increase the contrast of the lines, and the lines become more clear. Detection of these horizontal stripes in the spectrums of an image can help determine the quality of the image; so, the resulting image is converted to a binary image. From the binary image, the largest object (connected component) is selected. Ellipse will be plotted around the selected object. The length of the long diagonal of the ellipse is selected. When the length of the long diameter of the ellipse is larger than the threshold value (these values are determined by the experiment to be the threshold value that separates between clear and blurs images as shown in Table 1) then the image is blurred, otherwise, the image is clear. Figure 3 illustrates the steps of determining the image quality.

Table 1. The threshold value between clear and blurred images is according to image size.

Image size	Limit value
256 × 256	47.5
500 × 500	135
600 × 600	160
700 × 700	173.4

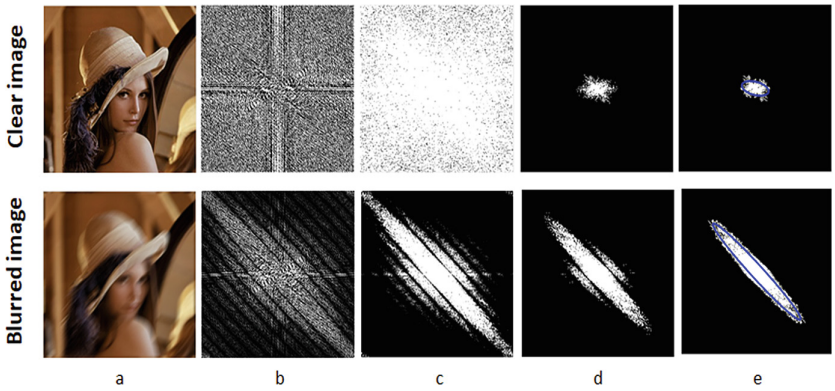

Fig. 3. (a) Authentic Image (b) frequency spectrum of an image (c) logarithmic transformation (d) Binary image (e) Extract the largest object and draw an ellipse.

6.2 Motion Blur Angle Estimation

One of the important parameters that are used in deblurring is motion blur angle. Steps for estimating motion blur angle are summarized in Algorithm 2.

Algorithm 2: Motion Blur Angle Estimation

Input: Blurred image (gray image)
Output: Blur angle

Begin
Step 1: Compute the Fourier transform F (u, v).
Step 2: Compute the log spectrum of F (u, v).
Step 3: Detect the edges of the image by using Sobel edge detection.
Step 4: Compute the Hough transform.
Step 5: Blur angle is the angle corresponding to the median of the largest five votes resulting from Hough peaks.
End.

The input image into Algorithm 2 is grayscale, the suggested method works efficiently with the grayscale images. The gray image will be transformed by using discrete Fourier transformation, in this step, we get spectrum horizontal lines (in dark, light, and white strips) which are perpendicular to the blur direction as shown in Fig. 4(c), but these lines are not clear, so, the logarithmic transformation is applied to the spectrum to increase the contrast of the lines, and the lines become more clear as shown in Fig. 4(d). Detection of these horizontal stripes in the spectrums of blurred images (created due to uniform linear motion) can help to estimate the blur angle.

The fourth step is the edge detection in the image resulting from step three by using the Sobel filter, the result is shown in Fig. 4(e), in this case, the image will be displayed as parallel lines (mostly discontinuous lines) around the sides of the spectrum center which called the top and bottom edges for the spectrum. So, to get continuous lines, the Hough transform will be applied on these edges to discover the direction of the line and determine the blur angle as shown in Fig. 4(f). The median of the five-largest votes will be selected from Hough peaks and the corresponding theta is the angle of blur direction.

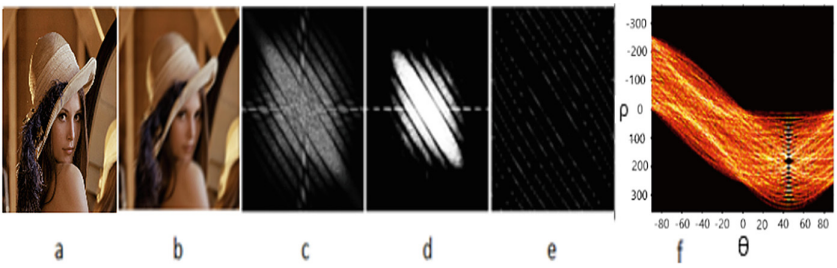

Fig. 4. (a) Authentic Image (b) Blurred image with angle 45 and length 15 (c) frequency spectrum of blurred image (d) logarithmic transformation (e) Sobel edge detection (f) Hough Transform

6.3 Motion Blur Length Estimation

The second essential parameter that is used in deblurring is motion blur length. The blur length estimation is based on the value of the blur angle estimated in Algorithm 2. The steps of estimating the blur length are summarized in Algorithm 3.

Algorithm 3: Motion Blur Length Estimation

Input: Blurred image (gray image), and motion blur angle.
Output: Blur length

Begin:
Step 1: Transform the blurred RGB image to a grayscale image.
Step 2: De-noising image based on the CCN denoising method.
Step 3: Compute the Fourier transform F (u, v).
Step 4: Compute the log spectrum of F (u, v).
Step 5: Performing inverse Fourier transform to get the image in the cepstrum domain.
Step 6: Rotate the image clockwise with the blur angle estimated from algorithm 2.
Step 7: Determine the average of all the columns of the image in step 6.
Step 8: Blur length is estimated by finding the lowest peak value.
End.

The input gray image will be denoised by using the convolution neural network (CNN) denoising method which works efficiently to remove unwanted particles. Then, the image is transformed into the cepstrum domain, because the cepstrum can be used to estimate the zero patterns in the frequency response from which one can estimate motion length. To transform an image into the cepstrum domain, first, transfer the image by Fourier transform and calculate logarithmic transformation for it. After that, the inverse Fourier transform was taken, which transform the image into the cepstrum domain. Next, rotate the image clockwise by the angle estimated in Algorithm 2 to make the blur direction perpendicular to the x-axis. The final step is to calculate the average of all the perpendicular lines which convert 2D images to one dimension vector. The blur length is estimated by finding the smallest peak values of the vector.

6.4 Create De-blurring Filter (PSF)

One of the most important de-blurring algorithms is generated in this section. This algorithm is a de-blur filter that works as the heart of the proposed system. The de-blur filter restores the original image using non-blind deconvolution, which is based on previously estimated blur angle and length. Steps for creating a de-blur filter are summarized in Algorithm 4.

Algorithm 4: Generate De-Blur Filter

Input: Length (***len***), and Angle (***theta***)

Output: De-Blur Filter

Begin

Step 1: Rotate half of the length estimated around the center, where

$$half = (len - 1)/2$$

Step 2: Convert the estimated angle from degree to radian.

Step 3: Compute sin and cos of the estimated angle.

Step 4: Compute the size of the half matrix according to sin and cos, where

$$sx = half * \cos(theta) + 1 - len * eps$$

$$sy = half * \sin(theta) + 1 - len * eps$$

Step 5: Construct two matrices according to sx and sy.

Step 6: Construct a new matrix based on step 5 by finding the shortest distance between each pixel location (i, j) and the rotating line, where the distance is perpendicular to the line, as shown below:

$$dist = y * \cos(theta) - x * \sin(theta)$$

Step 7: Find points beyond the endpoint of the line but within the width of the line, where

$$pix = (rad >= half) \& (abs(dist) \le 1)$$

$$rad = \sqrt{x\text{^}2 + y\text{^}2}$$

Step 8: Compute the distance between points outside the endpoint of the line and the endpoint of the line, where

$$xpix = half - abs((x(pix) + dist(pix) * sinphi)/cosphi)$$

$$dist(pix) = \sqrt{dist(pix).\text{^}2 + xpix.\text{^}2}$$

$$dist = 1 + eps - abs(dist)$$

Step 9: Convert the negative $dist$ value into zero as follows:

$$ist(dist < 0) = 0$$

Step 10: Rotate the matrix counter at a 180-degree angle clockwise.

Step 11: Convert the half-matrix to full size, where

$$H(end + (1:end) - 1,\ end + (1:end) - 1) = dist$$

Step 12: Normalization of the filter.

Step 13: Flip the upper rows with lower rows if the cosine of theta is greater than zero.

Step 14: De-Blur Filter has been created.

End.

The image can be divided into two halves. As a result, the line will cut the image into symmetric parts, whether horizontal, vertical, or diagonal. Therefore, the work will be based on the half-length of the line. The first step is to determine the size of the half-matrix based on the half-length, *cosine*, and *sine* of the blur angle. The size of the half-matrix will depend on the estimated angle and length in the previous algorithms. Two matrices are constructed depending on the size of the half-matrix calculated in the previous step. One of the matrices represents the horizontal line, and the other represents the vertical line. The orthogonal line is constructed based on these two matrices by finding the shortest distance between each pixel location and the rotating line, where the distance between these points and the line is perpendicular. To get an ideal line, first, find points far beyond the line's endpoint. The distance between these points and the line's endpoint is calculated. Finally, the values of these points are modified, and the matrix's values are updated according to the algorithms of the last two equations in step eight. After that, check whether the matrix values are less than zero, and update them to zero. Next, the matrix is rotated counterclockwise at 180° and modified to become a full-size matrix. The full-size matrix's values are normalized. Finally, the upper rows of the full-size matrix are flipped with the lower rows if the cosine of an angle is greater than zero. The line direction of the De-blur filter (Full-Size Matrix) is the same as the image's blur motion direction.

6.5 Deblured Image

Finally, non-blind Deconvolution techniques such as Wiener and Lucy-Richardson Filters were used to recover the sharp image using the PSF function created in the previous algorithm.

The following equation can be used to restore the original image using the Wiener filter:

$$\hat{F}(u, v) = W(u, v)G(u, v) \tag{4}$$

$\hat{\mathrm{F}}$(u, v) is the restored image, W(u, v) is the Wiener filter, and G(u, v) is the blurred image all in the frequency domain? The Wiener filter W(u, v) can be described as the equation below:

$$W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^{\wedge 2} + \frac{S\_U(u,v)}{S\_X(u,v)}} \tag{5}$$

where H(u, v) is the point spread function in the frequency domain, S_X is the signal power spectrum and S_U is the noise power spectrum. After the image was restored, it converted from the frequency domain to the spatial domain.

Also, the iterative Lucy-Richardson algorithm can be used to recover the original image by using the equation below:

$$\hat{f}_{k+1} = \hat{f}_k \left(h * \frac{g}{h \otimes \hat{f}_k} \right) \tag{6}$$

where $\hat{f}_{k+1}$ is the estimate of a restored image after k iterations, $*$ is the correlation operator, and h refers to the point spread function. The image $h \otimes \hat{f}_k$ is a blurred image.

7 Experimental Results and Discussion

In this proposal, the standard images such as Lena, Cameraman, Peppers, Baboon, Girl, Corridor, and Characters with different sizes of 256 × 256, 500 × 500, 600 × 600, 688 × 688, and 700 × 700 are selected to use in the evaluation the proposal efficiency under control environments. Images are blurred with different lengths and angles ranging from 10 to 85 pixels and 10 to 180°.

The first test was to detect if an image has blurred or not, a test implemented with different image sizes 256 × 256, 500 × 500, 600 × 600, and 700 × 700. The test was performed by using 1000 clear images from the VOC 2012 dataset. Also, blurred images were created by artificially blurring the images (fourteen standard and natural images were blurred with various lengths ranging "from 10 to 90 pixels", and various angles ranging "from 10 to 180°") to produce 8092 blurred images, in addition to fifty blurred images from (VOC 2012 dataset), that produces 8142 blurred images.

Figures 5, 6 and 7 illustrate the process of detecting a blur and the elapsed time required to determine whether an image has blurred or not for various image sizes. When images were blurry or clear, the proposed method produces good results for all image sizes.

Fig. 5. Implementation of blur detection algorithm on blurred images.

Mean absolute error (MAE) was calculated for blur detection in the blurry image; the value of MAE was very small, specifically for image sizes 256 × 256 and 700 × 700. In addition, when the image was clear, the MAE for the 500 × 500 and 600 × 600 image sizes has very small MAE, as shown in Table 2.

It is concluded that the detection of blur can be achieved very efficiently for different image sizes.

The second test was to measure the best image size that can be used to determine the blur angle. Blurred images with sizes 256 × 256, 500 × 500, and 600 × 600, and different blur lengths and angles are used in this test. The estimation of the angles for each image size is shown in Figs. 8, 9 and 10. Angles estimated when image size 256 × 256 gives good estimation when the blur length is from 10 to 60 but shows some error, specifically when the blur length is equal to (70, 80, and 85) which is eliminated when the image size is bigger.

Fig. 6. Implementation of blur detection algorithm on clear images.

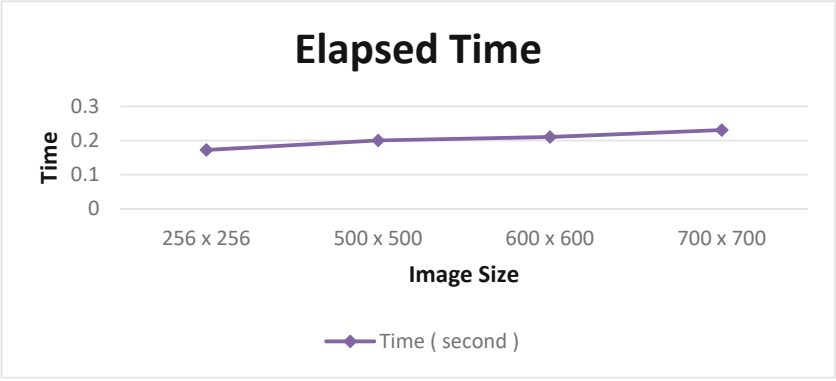

Fig. 7. Elapsed time to detect if an image has blurred or not for various image sizes

Table 2. Illustrate the MAE of the proposed method for different image sizes**.**

Image size	Number of clear images	Number of blurred images	MAE of Clear image	MAE of a blurred image
256 × 256	1000	8142	0.005	0.0001
500 × 500	1000	8142	0.003	0.001
600 × 600	1000	8142	0.003	0.001
700 × 700	1000	8142	0.009	0.0002

It is concluded from the testing results that the increasing image size will enhance the angle estimation specifically for blur lengths ranging from 15 to 85.

The errors that result when estimating blur angles for different length is shown in Fig. 11. The error is very small, one degree or less for just blur length equal to 10 and maybe a very small value with blur length equal to 15 and 25, while the others have no error.

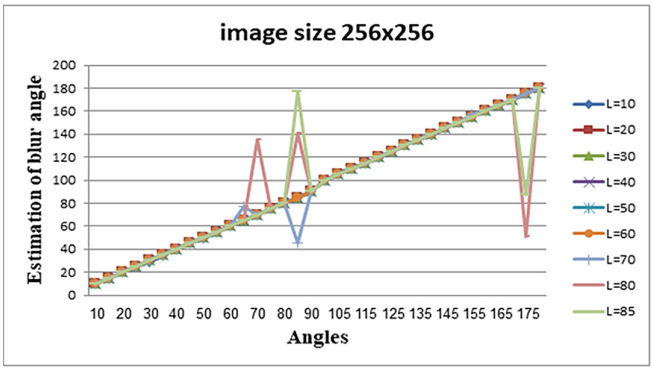

Fig. 8. Angle estimation for image size 256 × 256.

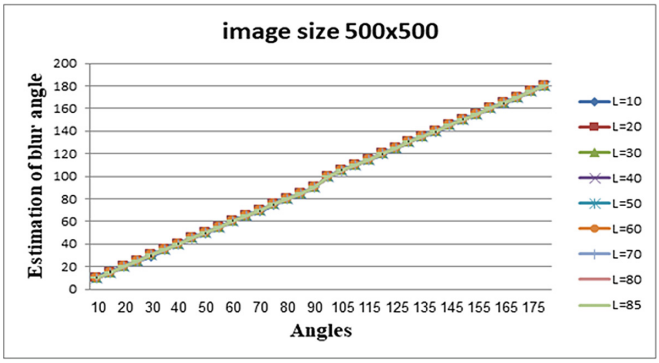

Fig. 9. Angle estimation for image size 500 × 500.

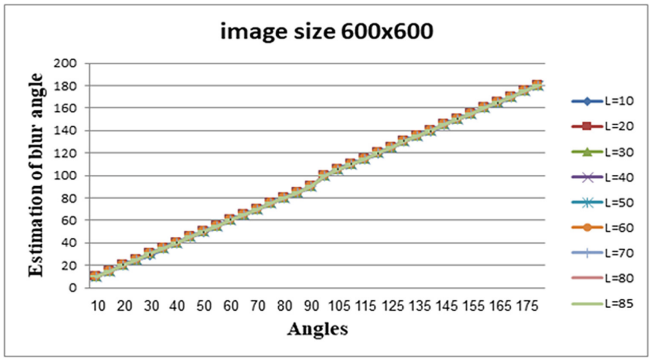

Fig. 10. Angle estimation for image size 600 × 600

Estimation of blur length for various blur angles to find the best image size that can give a more accurate estimation is the second test. The results from Fig. 12 approved

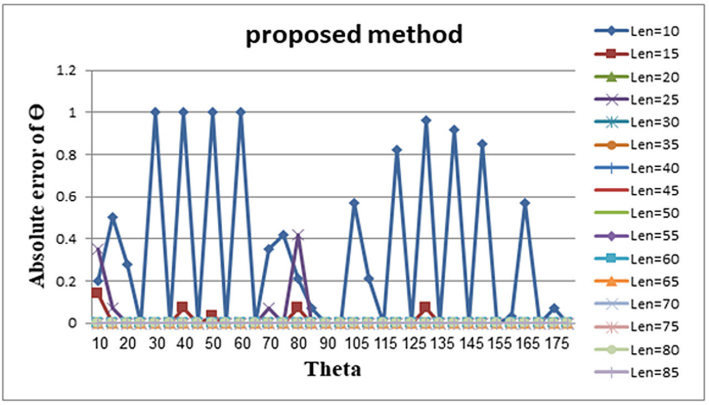

Fig. 11. Angle estimation errors for various blur length

that the small image size is not a good choice for estimating the length and direction of the blur. At the same time images with size 600 × 600 (Fig. 14) give some errors in the estimation of blur length for some images such as the cameraman image which gives a length of 11 instead of the real length of 15 when the angle was 40°. Also, the corridor image gives an outlier length when the angle equals 80. The best accuracy for length estimation was achieved when the image size is 500 × 500 as shown in Fig. 13.

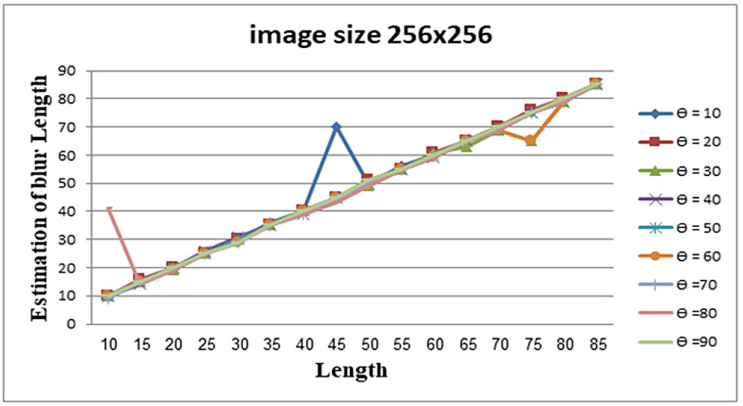

Fig. 12. Length estimation with different blur angles.

The errors that result when estimating blur length for different angles are shown in Fig. 15. The error is very small, less than one pixel mostly when the angles are 30 and 40.

We argue that the best accuracy can be achieved when the image size for this proposal is 500 × 500. At this size, there is a very good angle estimation and a very good length estimation.

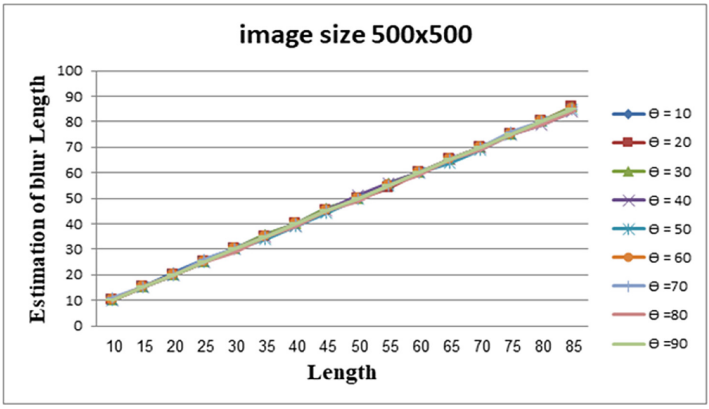

Fig. 13. Length estimation with different blur angles.

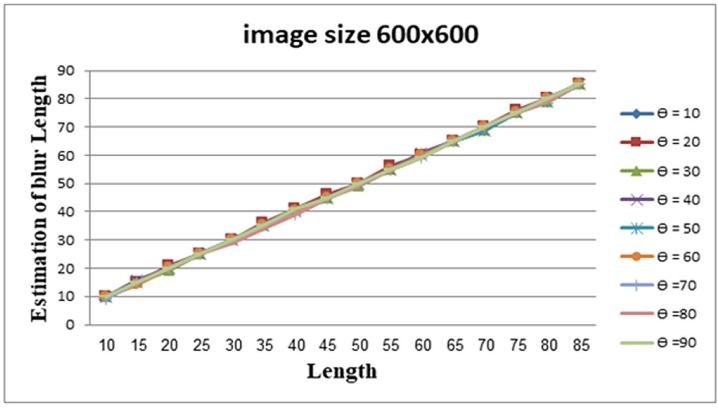

Fig. 14. Length estimation with different blur angles.

The other test is to measure the image quality after de-blurring. Different images with different lengths and angles were used in this test as shown in Fig. 16, the image size was 500 × 500, and the first row of images in Fig. 16 shows the blurred image, while the second row of images shows the de-blurred image after restoration by the proposed algorithm. The information about these images is summarized in Table 3. Table 3 shows the details about the PSF parameters used to blur images, while the last row shows the value of PSNR after image de-blurring.

Deblured images visually seem like pleasant images, where the values of PSNR encourage values.

Another test was to calculate image quality after blur was removed using Richardson-Lucy Filter. The test used eighty images from the VOC 2012 dataset entered into the system that was originally blurred at different lengths and angles, and the image size was 500 × 500, as shown in Fig. 17. Figure 17 shows the blurred image in the first row and

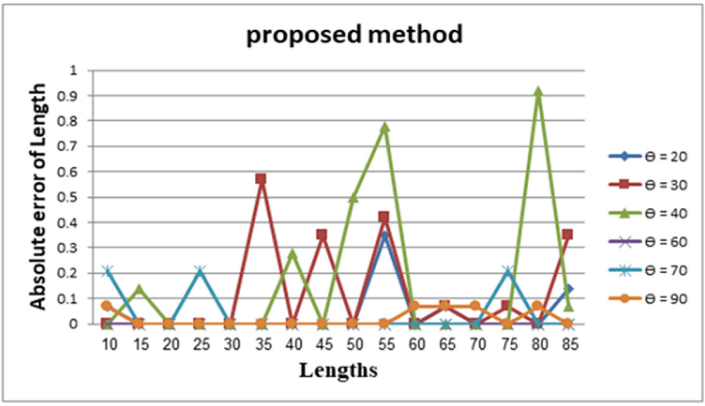

Fig. 15. Length estimation error for different angles.

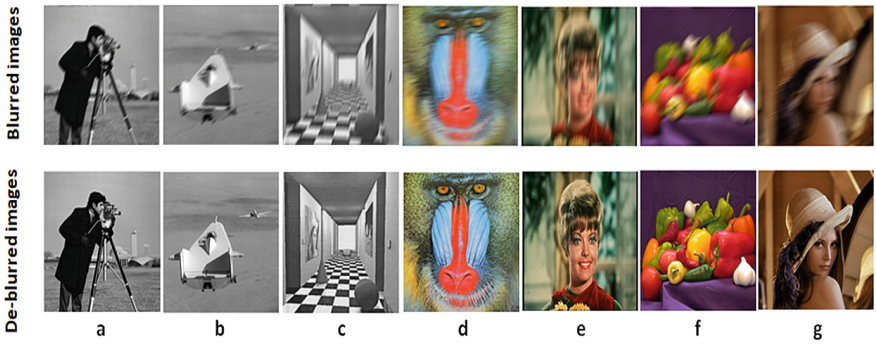

Fig. 16. Deblur images according to the proposed algorithm

Table 3. Information about the images in Fig. 16.

Images	A	b	c	d	e	F	g
Blur angle	45	30	70	10	80	40	20
Blur length	15	20	25	30	35	40	50
PSNR after de-blurring	111	112	103	101	113	114	109

the de-blurred image in the second row. After eliminating the blur, the Brisque values decreased, indicating that the image quality improved significantly.

The suggested blur angles estimated values, and run times are compared with other previous approaches such as Dash [13], Kumar [10], Wang [4], and Iraei methods [3], for different images with different sizes as shown in Table 4. From Table 4 it is concluded

Fig. 17. Deblur images according to the proposed algorithm using Richardson–Lucy Filter

that the result of the proposed method is more accurate than other estimation methods and the time is also the best.

Another comparison is listed in Table 5, which compared the suggested motion length estimation and the run time with the same methods used in comparing angle estimation. Different lengths for different image sizes are used in this comparison. The proposed method results achieve very good estimation, although some of the other previous methods produce comparatively good accuracy for length estimation than the proposed method.

Also, the absolute error of blur angles estimated shown in Fig. 11 are compared with other previous approaches such as Dash [13], Kumar [10], Wang [4], and Iraei methods [3].

The Dash method has a pretty good result to estimate angles smaller than $\theta = 45°$ for all blur lengths, in opposite for angles larger than $\theta = 45°$, where the absolute error rate is greatly increased. Kumar's method has estimated various angles as uniformly as possible rather than the two previously mentioned methods. This method still has a very high absolute error for $L = 10$, $L = 20$, and $L = 30$, and functionally is not suitable for these lengths. In the Wang method, there is a reasonably good approximation for all

Table 4. Results comparison of blur angle estimation using different methods.

Image	Original	Dash		Kumar		Wang		Iraei		Proposed Method	
	θ	θ	Time	θ	Time	θ	Time	θ	Time	θ	Time
	(°)	(°)	(s)	(°)	(s)	(°)	(s)	(°)	(s)	(°)	(s)
Lena 256x256	40	38.92	0.76	39.31	0.31	40.45	0.41	40.22	0.38	40	0.12
Cameraman 256x256	70	68.99	0.71	69.57	0.31	69.69	0.45	69.83	0.38	70	0.14
Pepper 256x256	10	9.11	0.79	10.65	0.31	10.4	0.42	9.76	0.39	10	0.126
Corridor 256x256	30	30.82	0.78	30.36	0.3	30.31	0.41	30.13	0.38	30	0.127
Building 600x600	35	33.61	1.91	33.94	1.09	35.41	1.37	35.39	1.4	35	0.129
Characters 688x688	60	61.62	2.32	60.76	1.34	59.61	1.86	59.62	1.82	60	0.135

Table 5. Results comparison of blur length estimation using different methods.

Image	Original	Dash		Kumar		Wang		Iraei		Proposed Method	
	L	L	Time	L	Time	L	Time	L	Time	L	Time
	(Pixel)	Pixel	(s)	Pixel	(s)	Pixel	(s)	Pixel	(s)	Pixel	(s)
Lena 256x256	30	28.88	0.62	30.37	0.34	29.69	0.44	30.16	0.64	30	2.37
Cameraman 256x256	20	21	0.6	20.41	0.34	19.71	0.43	20.26	0.63	20	2.40
Pepper 256x256	40	41	0.6	39.52	0.31	40.39	0.44	39.88	0.61	40	2.47
Corridor 256x256	50	49.55	0.61	49.61	0.32	49.64	0.41	50.17	0.62	50	2.31
Building 600x600	40	41.87	1.41	41.22	1.05	39.89	1.15	40.37	1.44	40	6.02
Characters 688x688	50	50.59	1.54	51.05	1.21	49.34	1.24	49.76	1.57	50	7.34

blur lengths at all angles, and the absolute error rate is reduced. Iraei method show high absolute error rate at L = 10, L = 20, and L = 30, but the error rate for some angles drops to less than 0.1 and has zero error when L = 50, L = 70, and L = 90. Also, the amount of this error will be reduced by increasing the blur angle. As it is clear in Fig. 11, the

suggested method that used Hough transform for edges detect, exhibit a great decrease of the absolute error rate for all angles and all blur length down to zero when the length of motion L = 20, L = 30, L = 40, L = 50, L = 60, L = 70, L = 80 and L = 85 for all angles.

Figure 15 shows the absolute error rate of the proposed method when blurred with blur lengths from 10–85 (with step 5 pixels) for six different blur angles. Figure 15 also shows that the proposed method has a maximum absolute error of 0.9 in the worst case. Most lengths have an absolute error of less than 0.3, which means restoring the blurry image is more accurate than other methods.

Another test is implemented on noisy blurred images to prove the robustness of the de-blurring method to noise. Gaussian noise was added to eight different images with different sizes of signal-to-noise ratio (SNR = 25 dB). The five methods (Dash's, Kumar, Wang, Iraei, and the proposed methods) are used to estimate the PSF parameters, and then reconstruct the original images. The results showed in Table 6. As Table 6 shows, the maximum absolute error for the estimated blur angle by the proposed method is reached zero, and it is a very good result compared with other methods, although the results of other methods were very good in general. Same thing for the blur length the results were very good except for a small deviation of estimation lengths for some images.

Table 6. Results comparison of blur parameters estimation in noisy states (SNR = 25 dB)

Image	Original		Dash			Kumar			Wang			Iraei			Proposed Method		
	θ	L	θ	L	PSNR	θ	L	PSNR	θ	L	PSNR	θ	L	PSNR	θ	L	PSNR
Lena 256x256	40	30	38.71	28.7	25.29	40.48	29.2	31.22	39.59	29.57	37.16	40.27	29.84	44.03	40	30	99.14
Cameraman 256x256	70	20	68.91	21	25.46	70.46	19.2	30.12	69.62	19.69	36.8	69.71	20.31	43.36	70	20	100.95
Pepper 256x256	10	40	9.03	39.04	25.07	9.33	40.5	30.09	10.41	40.42	36.27	9.73	39.79	42.88	10	40	93.63
Corridor 256x256	30	50	28.94	49.15	26.39	29.29	49.5	31.54	29.57	49.59	36.73	29.8	50.22	43.46	30	50	94.59
Baboon 256x256	40	60	39.09	61	27.1	39.24	60.8	31.87	39.6	59.57	37.86	40.23	60.17	44.68	40	60	91.04
Girl2 256x256	20	85	21.06	85.45	27.24	19.64	84.6	31.98	20.28	85.39	37.35	19.79	85.29	44.19	20	85	90.78
Building 600x600	35	40	32.84	42.03	24.74	34.06	41.1	30.56	35.04	39.72	36.07	35.56	40.34	42.45	35	40	99.77
Characters 688x688	60	50	61.73	51.6	25.51	60.88	51	30.64	39.63	49.4	35.47	59.78	49.75	41.97	60	50	101

Moreover, the PSNR of de-blurred noise was good in general, and it reflects the good robustness of the noise.

8 Conclusion

In this work, the deblur image restoration is constructed by determining the PSF. The estimation of angle and length by the proposed method gives more accurate results than the other methods. The main contribution of this work is the ability to check the blurred image before the de-blurring process. The range of estimation angles and length is wider than the previous methods. In this work, we prove that size of the image will have significant effects on the accuracy of estimation of the PSF, and then restore the de-blurring image. Generally, the absolute errors are reduced. This method is robust and works very well even with noisy images. The suggested method produces a pleasant deblur image and it is reliable.

References

1. Dawood, A.A.M., Saleh, M.F.: Image deblurring techniques: a review. J. Sci. Eng. Res. **6**(3), 94–98 (2019)
2. Gowthami, S., Harikumar, R.: Conventional neural network for blind image blur correction using latent semantics. Soft. Comput. **24**(20), 15223–15237 (2020). https://doi.org/10.1007/s00500-020-04859-y
3. Iraei, I., Sharifi, M., Baleghi, Y.: A new approach to enhancing the estimation of blur parameters in blurred images. Optik **224**, 165298 (2020). https://doi.org/10.1016/j.ijleo.2020.165298
4. Wang, Z., Yao, Z., Wang, Q.: Improved scheme of estimating motion blur parameters for image restoration. Digit. Signal Process. A Rev. J. **65**, 11–18 (2017). https://doi.org/10.1016/j.dsp.2017.02.010
5. Li, N., Chen, S., Tang, M., Kan, J.: Deblurring method for motion blurred images based on GAN. Int. J. Circ. Syst. Signal Process. **15**, 106–113 (2021). https://doi.org/10.46300/9106.2021.15.12
6. Zhang, G., Zheng, T.: Estimation method of blurred parameters in moving blurred image. J. Phys. Conf. Ser. **1616**(1) (2020). https://doi.org/10.1088/1742-6596/1616/1/012096
7. Elmi, Y., Zargari, F., Rahmani, A.M.: Iterative approach for parametric PSF estimation. Multimedia Tools Appl. **79**(39–40), 29433–29450 (2020). https://doi.org/10.1007/s11042-020-09511-3
8. Murthy, K.V.V., Gajjar, R., Zaveri, T., Banerjee, A.: Trigonometry-based motion blur parameter estimation algorithm. Int. J. Image Min. **3**(1), 67 (2018). https://doi.org/10.1504/ijim.2018.10014060
9. Albluwi, F., Krylov, V.A., Dahyot, R.: Image deblurring and super-resolution using deep convolutional neural networks, pp. 1–6 (2018). https://doi.org/10.1109/MLSP.2018.8516983
10. Kumar, A.: Deblurring of motion-blurred images using histogram of oriented gradients and geometric moments. Signal Process. Image Commun. **55**, 55–65 (2017). https://doi.org/10.1016/j.image.2017.03.016
11. Win, Z.M., Aye, N.: Detecting image spam based on file properties, histogram, and hough transform. J. Adv. Comput. Netw. **2**(4), 287–292 (2014). https://doi.org/10.7763/jacn.2014.v2.127
12. Park, J., Kim, M., Chang, S., Lee, K.H.: Estimation of motion blur parameters using cepstrum analysis. In: Proceedings of International Symposium on Consumer Electronics, ISCE, pp. 406–409 (2011). https://doi.org/10.1109/ISCE.2011.5973859
13. Dash, R., Majhi, B.: Motion blur parameters estimation for image restoration. Optik (Stuttg). **125**(5), 1634–1640 (2014). https://doi.org/10.1016/j.ijleo.2013.09.026