



Detection of local motion blurred/non-blurred regions in an image

B R Kapuriya¹ · Debasish Pradhan² · Reena Sharma³

Received: 27 July 2021 / Revised: 7 December 2021 / Accepted: 29 September 2023 /

Published online: 23 October 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Motion blur of an image is a common phenomenon that occurs while taking a photograph due to the relative movement of the object and an image acquiring device. It is essential to detect this phenomenon of blurring of images in many applications such as information retrieval. This paper proposes a novel local blur detection technique, and it performs better than the existing works. This technique mainly uses Radon transform and Laplacian of Gaussian on the local neighborhood around each pixel to estimate blur information. Additionally, two new weight functions are introduced based on local geodesic distance and local variance. It is shown that these functions play a significant role in segregating blur and non-blurred parts. Simulation results validate the correctness and accuracy by testing the proposed algorithm on some challenging images with similar color information in the foreground and background. Various quantitative performance measures have determined the superiority of the proposed method.

Keywords Fourier transform (FT) · Geodesic distance · Local blur angle estimator (LBAE) · Point spread function (PSF) · Laplacian of gaussian (LoG) · Radon transform (RT)

1 Introduction

It is well known that blurred images are generated due to either motion of the object or due to the motion of the camera while capturing the image. Therefore two types of blurred

✉ Debasish Pradhan
pradhandeb@gmail.com

B R Kapuriya
kapuriyabr@gmail.com

Reena Sharma
reena@cabs.drdo.in

¹ Centre for Airborne Systems (CABS), Bangalore 560037, India

² School of Computer Engineering and Mathematical Sciences, Defence Institute of Advanced Technology (DIAT), Pune 411025, India

³ Aeronautics Research and Development Board (ARDB), DRDO, Delhi 110054, India

images exist, i.e., 1) partial blur and 2) global blur. In both cases, researchers have developed numerous techniques for their detection and restoration. Many researchers have discussed the restoration of global blur images, which are presented in [4, 5, 7, 15, 16, 24, 26, 29, 31, 34, 35, 39]. Most of these techniques have followed the well-known blur model for linear motion, which is defined as follows:

$$g(\mathbf{z}) = h(\mathbf{z}) \otimes f(\mathbf{z}) + n(\mathbf{z}), \quad (1)$$

where $g(\mathbf{z})$ is an observed image, $h(\mathbf{z})$ is a degraded function, $f(\mathbf{z})$ is an original image, and $n(\mathbf{z})$ is a noise function. And " \otimes " is two dimensional convolution operator. Here, we denote $\mathbf{z} = (x, y)$ as 2 dimensional domain of an image and $d\mathbf{z} = dx dy$ as suitable integral. This notation will be followed throughout this paper. In this model, the degraded function produces cumulative effects of distortions in the full image after convolution with the original image.

In the case of a partial blur, one or more objects are blurred during photography. Two types of blur models are shown in Fig. 1(a) & (b). Here, we focus on one or more linear motions of an object in the image. To restore this type of blurred object, one needs to detect the blurred object and subsequently apply blur restoration techniques. Many of these techniques are studied in [8, 9, 11, 19, 22, 27, 28, 36, 40]. But the challenge lies in detecting exact blurred regions of the degraded image. Hence, we have focused on detecting precise blurring regions of an image. Several techniques have been proposed in the literature to detect & segment partial blur in the image. Here, we have briefly reviewed a few of them. Pan et al. [14] have used maximum a posteriori framework to segment and deblur various blurred regions in an image. In [1], Chakrabarti et al. have introduced a likelihood measure using local Fourier Transform to determine whether a specific kernel blurs a particular image patch or not. Liu et al. [23] have proposed region-based features, namely, local power spectrum slope, gradient histogram span, and maximum saturation for identifying blur regions. In addition, they have discussed local autocorrelation congruency features for classifying detected blur either in motion or defocus blur. Similarly, Kalalembang et al. [17] have used block-based discrete cosine transform for blur detection, and Tang et al. [37] used log averaged spectrum residual

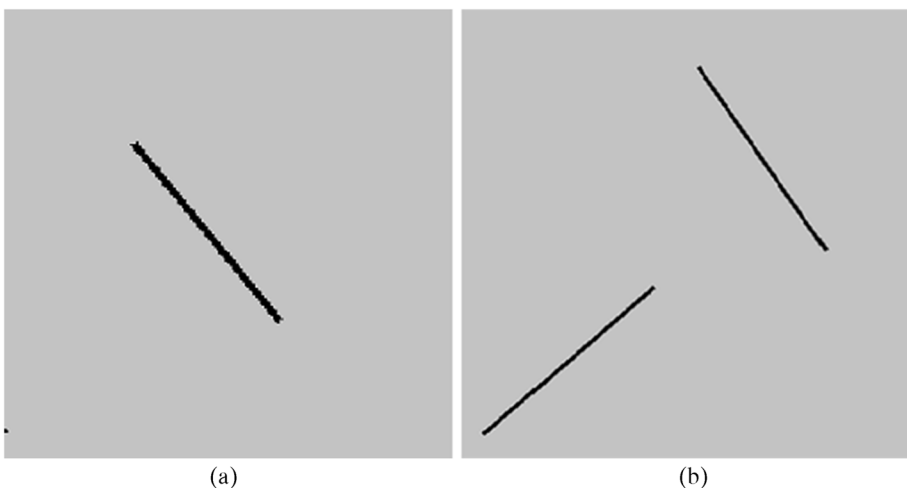


Fig. 1 Different kind of Point Spread Functions (PSF): (a) Linear motion PSF (b) Two linear motions in different directions

to estimate blur map. In [32], Shi et al. have modified the set of blur features which are derived from multiple domains such as gradient histogram, kurtosis, and local filters, for blur detection. In [2], Bahrami et al. have estimated local blur kernels from gradient images, using an image patch, which generates a relative blur degree for each patch to classify it as blur or non blur patch. Similarly, in [30], Paramanand et al. have used a local image patch to estimate the transformation spread function around each pixel to classify it as blurred or non-blurred pixels by manually selecting the patches. In [3], Bini et al. have proposed PDE based image deblurring method wherein they used diffusive image smoothing term and a reactive image enhancement term to construct the nonlinear level set model. Devy et al. [8] proposed a local blur kernel estimator based on two filters, i.e., Gabor and dictionary, to study the pattern around each pixel. However, the number of filters is decided manually, which limits the scope of this method. In [33], Su et al. have detected blur by examining the singular value information for each image pixel, and subsequently, alpha channel is used to classify the type of the blur. In [10], Golestaneh et al. have proposed a method to estimate spatially varying blur from a single image by using a multi-scale transform decomposition followed by fusion and sorting of high-frequency coefficients of gradient magnitudes. This method detects both defocus and motion blur without prior knowledge of blur type. Similarly, one of the effective technique described by Javaran et al. in [13] also detect and segment blur and nonblur region for both types of a blur - defocus blur and motion blur. In the case of real images, the techniques discussed so far may not produce connected blur/non blur regions. Recently, few deep learning-based methods have been explored. Kim et al. [20] used a deep convolutional neural network for the detection of defocus and motion blur technique. Subsequently, Ma et al. [25] have used a support vector machine (SVM) to classify image patches in a blur and non-blur regions after training Deep Neural Network (DNN). Other DNN based research works can be found in [12, 38]. These techniques are capable of detecting motion blur regions up to a certain accuracy. There is still a need for further improvement in detecting the exact portion of blurred regions. In this paper, we propose a novel approach to detect local blur in an image where blurred and non-blurred regions have similar colors. In the existing literature, this issue has been addressed partially. In our approach, we applied Radon transform and Laplacian of Gaussian on a local neighborhood around each pixel in the image and observed that the values of these Radon transform give significant differences for blurred and non-blurred pixels. We utilized this phenomenon along with the work proposed in [18] to classify each pixel either in blurred or non-blurred pixels. The main contributions of this paper are listed below:

- A novel local blur estimator is proposed based on the value of Radon transform (RT) of LoG and local blur angle estimator (LBAE). Here, it is emphasized that a local blur estimator is used for calculating blur information, while a local blur angle estimator is used for finding a blur angle at each pixel in an image. The main framework of the proposed method is shown in Fig. 2.
- Two new weight functions $k_1(z)$ and $k_2(z)$ have been suggested, which play a significant role in deciding pixel-wise weightage of blur information derived from LBAE and the value of RT of LoG, respectively.
- Few variants of the second weight function $k_2(z)$ are proposed, and their performance analyzed.

The outline of this paper is as follows. In Section 1, we briefly discussed previous works related to the detection and segmentation of partial blur. In Section 2, we have reviewed the concepts of blur model, Radon transform, Laplacian of Gaussian, and local blur angle estimator, which are used extensively in the proposed method. A novel local blur estimator along

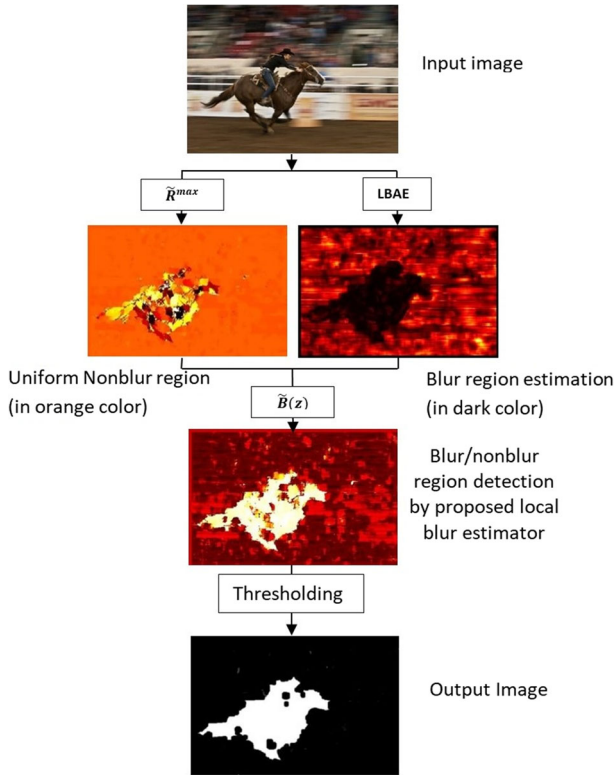


Fig. 2 Framework of the proposed technique

with two weight functions are proposed in Section 3. Section 4 describes the experimental results of the proposed model and evaluation of the quantitative performance of the proposed method with other existing methods. Conclusions are discussed in Section 5.

2 Background

In this section, we present the concepts of the blur model, Radon transform, Laplacian of Gaussian, directional derivative and earlier works of local blur angle estimation technique. In this section, we present the concepts of the blur model, Radon transform, Laplacian of Gaussian, directional derivative, and earlier works of local blur angle estimation technique.

2.1 Blur model

Blur kernel of one directional motion blur model [6] is described as:

$$K^L = \begin{cases} \frac{1}{L}; & \text{if } \sqrt{x^2 + y^2} \leq \frac{L}{2} \text{ and } \frac{y}{x} = \tan\theta, \\ 0; & \text{otherwise,} \end{cases} \tag{2}$$

where, L is a blur length, θ is a blur angle, and (x, y) is pixel position. Many techniques are used for estimating blur parameters. Primarily used estimation is mentioned in [21]. The argument related to the maximum value of Radon transform (RT) gives blur angle, and the motion blur length is estimated using the cepstrum of a blurred image which is the distance between two significant negative peaks from the origin [21].

The original image can be restored using a convolution theorem, provided the blur kernel is known. The linear motion blur model described in (2), will work in the case of a partial blur, only if the object is uniformly moving in a linear direction. But, this model fails to deblur an object without finding a proper blur region. Therefore, it is essential to detect the blurred region for applying further post-processing techniques.

2.2 Operators: RT, LoG, directional derivative

The Radon transform (RT) is used to measure the strength of a line in an image, the details of RT are well discussed in [29] and is defined as:

$$R(I, \rho, \theta) = \int_{-\infty}^{\infty} I(\rho \cos \theta - y \sin \theta, \rho \sin \theta + y \cos \theta) dy, \quad (3)$$

where $R(I, \rho, \theta)$ denotes Radon transform and it is an integral in an image I along the line at an angle θ and at distance ρ from origin. Laplacian of Gaussian (LoG) is used for sharpening the edges after suppressing the noise and it is defined as:

$$LoG = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \quad (4)$$

where σ is a standard deviation, x and y are pixel locations. Directional derivative of an image I at pixel location (a, b) in the direction of vector $v = (v_1, v_2)$ is defined as

$$\nabla_v I = \nabla_x I \cdot v_1 + \nabla_y I \cdot v_2, \quad (5)$$

where ∇_x and ∇_y are derivative operators.

2.3 Local blur angle estimation

In this subsection, we have discussed the local blur angle estimation method in detail, which is mentioned in [18]. Here, the authors have introduced a local blur angle estimator (LBAE) to find pixel-wise blur angle throughout an image using LoG and RT, and it is defined as

$$\tilde{\theta}^l = \text{Arg} \max_{\theta} \text{Var} \left\{ \nabla_y \left[R \left(LoG(\nabla_u I) \right) \right] \right\}, \quad (6)$$

where $\tilde{\theta}^l$ denotes estimated blur angle for each pixel, l indicates local blur angle and $R(LoG(\nabla_u I))$ is the composition of operators where R , LoG and ∇_u are denoted as Radon transform, Laplacian of Gaussian and directional derivative, respectively. All these operators are applied sequentially on a small window that is selected around each pixel in an image. Here, a directional derivative operator is used for sharpening the edges. To use a directional derivative, the availability of prior information about the likely blur direction is essential. The prior information of likely blur direction is helpful in increasing the accuracy of blur

angle estimation. Blur angle estimators have been proposed for finding likely blur directions, in [18], which are defined as follows:

$$\hat{\theta}_k^g = Arg \max_{\theta \neq \hat{\theta}_1^g, \hat{\theta}_2^g, \dots, \hat{\theta}_{k-1}^g} Var \left\{ \nabla_y \left[R(\nabla I) \right] \right\}, \tag{7}$$

where $\hat{\theta}_k^g$, $k = 1, 2, \dots, n$'s are likely blur angle existing in an image related to n different objects and g indicates for global blur angle. In the case of the non-availability of likely blur directions, one can use a normal gradient operator.

Subsequently, the LoG is applied after applying directional derivative followed by RT. The index of the maximum value of RT gives a blur angle. This estimator determines the blur angle effectively from objects blurred by different blur parameters. However, it fails in segregating blurred and non-blurred regions accurately, which can be seen from Fig. 9(b) for the erroneous outcomes in the non-blurred region. In this paper, we have proposed a new technique to overcome this problem.

3 Proposed technique

In this section, we have described a novel local blur estimator along with two weight functions. The goal of this method is to detect blurred and non-blurred regions accurately and efficiently. The proposed method primarily uses LoG and RT on a small window in two different estimators. The proposed local blur estimator is defined as

$$\tilde{B}(\mathbf{z}) = k_1(\mathbf{z})\tilde{\theta}^l(\mathbf{z}) + k_2(\mathbf{z})\tilde{R}^{max}(\mathbf{z}), \tag{8}$$

where $\tilde{\theta}^l(\mathbf{z})$ is

$$\tilde{\theta}^l(\mathbf{z}) = Arg \max_{\theta} Var \left\{ \nabla_y \left[R(LoG(\nabla I(\mathbf{z}))) \right] \right\}. \tag{9}$$

Here, $\tilde{\theta}^l$ is locally estimated for each pixel and other parameters in (9) are the same as in (6). Note that the (9) and (6) are the same if likely blur directions in (6) are not considered, and it is used for emphasizing blur region detection. In the second term, $\tilde{R}^{max}(\mathbf{z})$ is used to emphasize non-blur region and is defined as

$$\tilde{R}^{max}(\mathbf{z}) = \max_{\theta} Var \left\{ \nabla_y \left[R(LoG(\nabla I(\mathbf{z}))) \right] \right\}. \tag{10}$$

Here, $\tilde{R}^{max}(\mathbf{z})$ is the maximum value of $R(LoG(\nabla))$ in the neighbourhood of each pixel in $I(\mathbf{z})$, where R and LoG are defined in (3) & (4), respectively. In each window of an image, RT calculates the strength of the lines at a particular angle. Therefore, the value of $\tilde{R}^{max}(\mathbf{z})$ is higher at the blurred region points as compared to non-blurred region points. Experimentally, it is observed that in the non-blurred region, the $\tilde{R}^{max}(\mathbf{z})$ is consistently lower, as also seen from Fig. 3(b).

The weight functions are defined as

$$k_1(\mathbf{z}) = \frac{1}{1 + \beta(\tilde{\theta}^l(\mathbf{z}) - M_{\tilde{\theta}})^2} \text{ and } k_2(\mathbf{z}) = \frac{1}{1 + \alpha d_l^{ge}(\mathbf{z})}. \tag{11}$$

Here, $M_{\tilde{\theta}}$ is mode of $\tilde{\theta}^l(\mathbf{z})$ and $d_l^{ge}(\mathbf{z})$ is a maximum value of edge weighted geodesic distance which is calculated locally from $\tilde{R}^{max}(\mathbf{z})$ & (12).

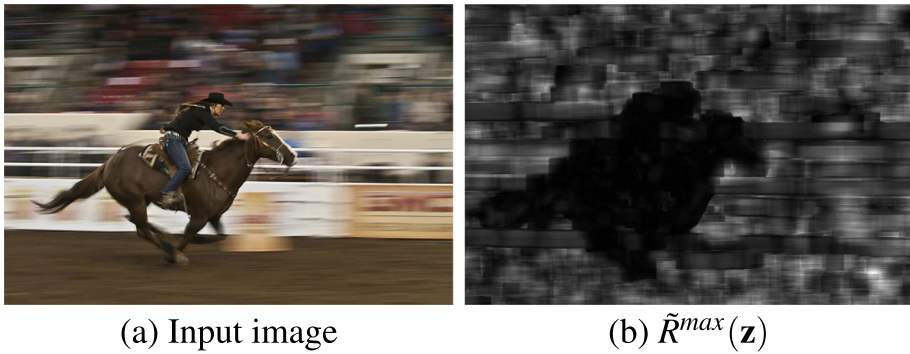


Fig. 3 a) Test image: Horse b) Corresponding $\tilde{R}^{max}(\mathbf{z})$ calculated by taking 12×12 window size

The constants α and β are used for generating weight of the function $k_2(\mathbf{z})$ and $k_1(\mathbf{z})$, respectively. In particular, the smaller value of these constants implies giving more weightage to the corresponding function. The values of these constants taken for various test images are recorded in Table 3.

The first term in (8) provides more weightage to the blurred region of an image. The weight function $k_1(\mathbf{z})$ is selected in such a way that it vanishes at the non-blurred region and approaches to 1 at the blurred region as depicted in Fig. 4(b). Whereas the second term of (8) provides more weightage to the non-blurred region of an image. The weight function $k_2(\mathbf{z})$ is selected in such a way that it vanishes in blurred region and approaches to 1 in non-blurred region, which is opposite of $k_1(\mathbf{z})$ and, can be seen in Fig. 4(c).

In each local window $I_B(\mathbf{z})$, geodesic distance is calculated for each pixel from a central candidate pixel (x_0, y_0) by $d_l^{ge}(\mathbf{z}) = 0$ for $\mathbf{z} = (x_0, y_0)$ and $d_l^{ge}(\mathbf{z}) = \frac{d_B(\mathbf{z})}{\|d_B(\mathbf{z})\|_{L^\infty}}$ for other

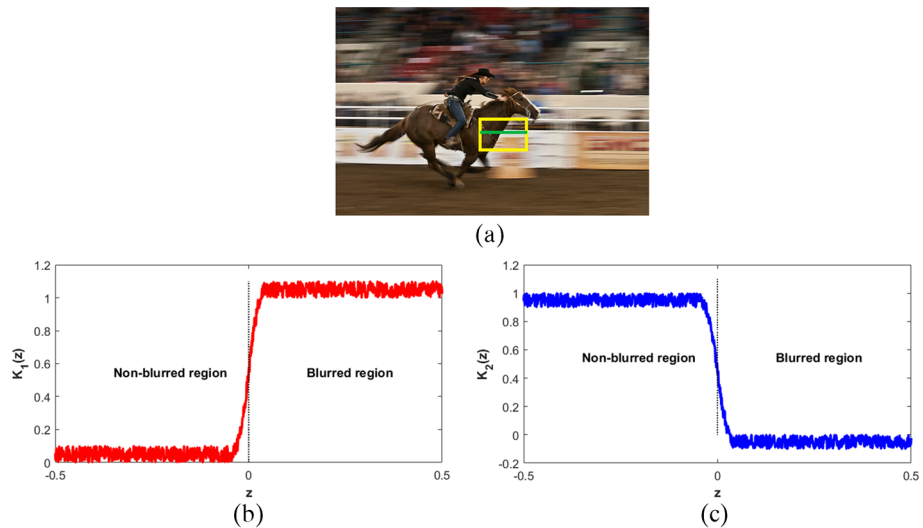


Fig. 4 (a) Test image: Horse-500×332 (b) The values of the function $k_1(z)$ along the green line inside the rectangle (superimposed as yellow color) (c) The value of the function $k_2(z)$ along the green line inside the rectangle (superimposed as yellow color)

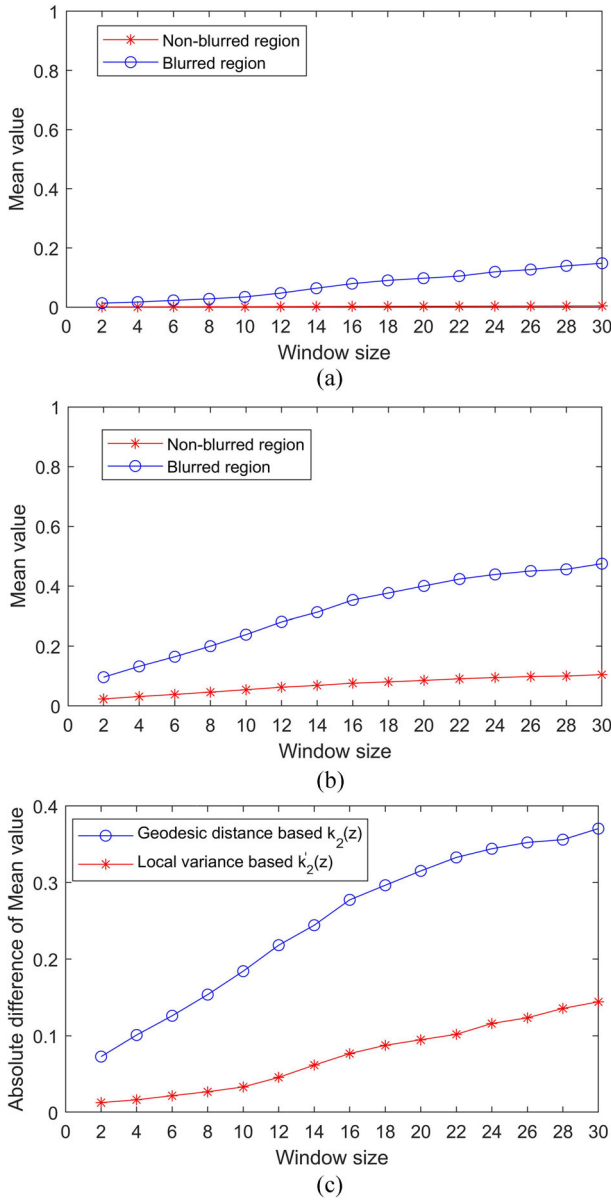


Fig. 5 (a) The average value of $k_2'(z)$ derived using local variance (b) Average value of $k_2(z)$ derived based on geodesic distance function (c) Absolute difference between blurred and non-blurred region. Number of samples: 1000 random pixels from each region. Test image shown in Fig. 4(a) has used for computation

pixels \mathbf{z} in B , where $d_B(\mathbf{z})$ is determined by solving the following PDE:

$$\begin{aligned}
 |\nabla d_B(\mathbf{z})| &= f(\mathbf{z}), \\
 d_B(\mathbf{z}) &= 0, \mathbf{z} \in B.
 \end{aligned}
 \tag{12}$$

Here, $f(\mathbf{z}) = \epsilon_g + \beta_g | \nabla G_\sigma \otimes I_B(\mathbf{z}) |$, where ϵ_g and β_g are constant parameters and G_σ is a Gaussian function with standard deviation σ to make the $I_B(\mathbf{z})$ free from noise and \otimes is a convolution operator. Here, $f(z)$ is a distance function which calculate distance between central pixel (i.e., $\mathbf{z} = (x_0, y_0)$) and other pixels in the small window. This distance function $f(\mathbf{z}) = \epsilon_g + \beta_g | \nabla G_\sigma \otimes I_B(\mathbf{z}) |$ is fulfilling the properties, where if $f(z)$ is small then we have flat region and if $f(z)$ is large then we have edgy region. And keeping the values of the constants $\epsilon_g = 10^{-3}$ and $\beta_g = 1000$, we get small distance in the flat region because $| \nabla G_\sigma \otimes I_B(\mathbf{z}) | = 0$ in the flat region, whereas $| \nabla G_\sigma \otimes I_B(\mathbf{z}) |$ will be large in edgy region. Other choices of $k_2(\mathbf{z})$ are

$$k'_2(\mathbf{z}) = \frac{1}{1 + \alpha_1 Var^l(\mathbf{z})} \quad \text{and} \quad k''_2(\mathbf{z}) = \frac{1}{1 + \alpha_2 H(\mathbf{z})}, \tag{13}$$

where $Var^l(\mathbf{z})$ is local variance in \tilde{R}^{max} for each pixel and H is the Heaviside function which is defined as

$$H(\mathbf{z}) = \begin{cases} 0; & \tilde{R}^{max}(\mathbf{z}) < \epsilon \text{ for non-blurred region} \\ 1; & \tilde{R}^{max}(\mathbf{z}) > \epsilon \text{ for blurred region.} \end{cases}$$

Here, $\epsilon > 0$ needs to be chosen manually dependent on $\tilde{R}^{max}(\mathbf{z})$ value, which varies from image to image. For instance, if the object is motion blurred with a small blur length, $\tilde{R}^{max}(\mathbf{z})$ values of blur and non-blur region has narrow difference. In this way, choice of manual $\epsilon > 0$ became significant and picked so that a large portion of the pixels from non-blur region have $\tilde{R}^{max}(\mathbf{z}) < \epsilon$. In view of investigations, we observed that the $k_2(\mathbf{z})$ is more reliable, however $k'_2(\mathbf{z})$ has a superior computational proficiency. While $k''_2(\mathbf{z})$ is more reliable and efficient if the value of ϵ is picked appropriately. For automatic computation, one has to choose either $k_2(\mathbf{z})$ or $k'_2(\mathbf{z})$. By comparing the mean values of $k_2(\mathbf{z})$ for blurred and non-blurred region as shown in Fig. 5 (b), it is seen that the difference is more for $k_2(\mathbf{z})$ in blurred and non-blurred region. By similar comparison for $k'_2(\mathbf{z})$, it is seen that $k'_2(\mathbf{z})$ produces a narrow gap for blurred and non-blurred region which is shown in Fig. 5 (a). This is depicted in Fig. 5(c) in terms of absolute differences in blurred and non-blurred region of $k_2(\mathbf{z})$ and $k'_2(\mathbf{z})$. Therefore, we conclude that geodesic distance based function matches better to the desired function as compared to local variance based function as seen in Fig. 5(c). Pseudo code of the proposed method is presented in algorithm 1.

Algorithm 1 Pseudo code of the proposed technique.

- 1: Input: Blurred image I
 - 2: Steps to calculate $\tilde{R}^{max}(\mathbf{z})$ and $\tilde{\theta}^l(\mathbf{z})$;
 - 3: Consider small neighbourhood $I_s(z)$ around each pixel $I(z)$
 - 4: Apply gradient operator ∇ on sub-image $I_s(z)$
 - 5: Apply LoG on resultant outcome
 - 6: Determine R by Applying 'Radon transform' after LoG
 - 7: Apply vertical oriented directional gradient operator on R
 - 8: Calculate $\tilde{\theta}^l(\mathbf{z})$ using (9) and $\tilde{R}^{max}(\mathbf{z})$ using (10)
 - 9: Find $k_1(\mathbf{z})$ and $k_2(\mathbf{z})$, for each (i, j) , using (11)
 - 10: Find $\tilde{B}(\mathbf{z})$ using (8)
 - 11: Output: Segmented image
-



(a) Test image 1



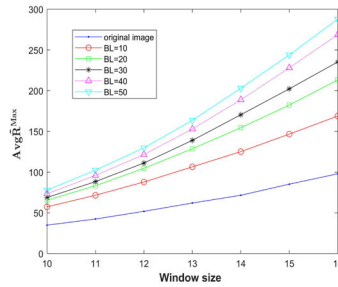
(b) Blur image at 10° blur angle



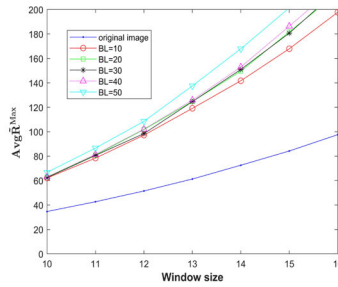
(d) Blur image at 30° blur angle



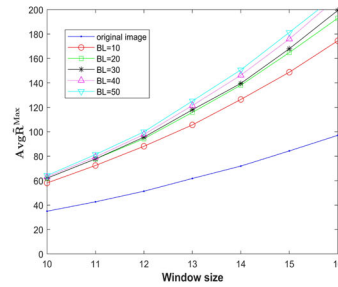
(f) Blur image at 60° blur angle



(c) Comparison of \bar{R}^{max} for 10° blur angle

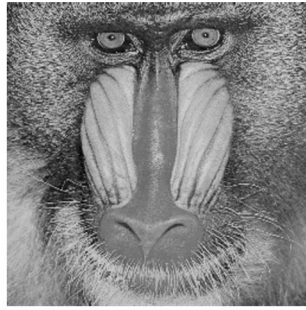


(e) Comparison of \bar{R}^{max} for 30° blur angle



(g) Comparison of \bar{R}^{max} for 60° blur angle

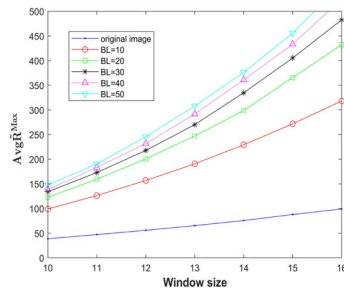
Fig. 6 (a) Test image: Lena-512×512. (b), (d) & (f) Blur images with 10°, 30° and 60° blur angle and 50 pixels blur length, respectively. (c),(e) & (g) Comparison of the average value of \bar{R}^{max} for various blur lengths- 10, 20, 30, 40, and 50 pixels. Here note that BL means blur length



(a) Test image 2



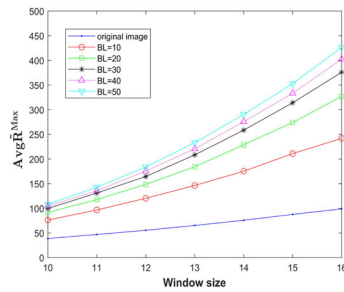
(b) Blur image at 10° blur angle



(c) Comparison of \bar{R}^{max} for 10° blur angle



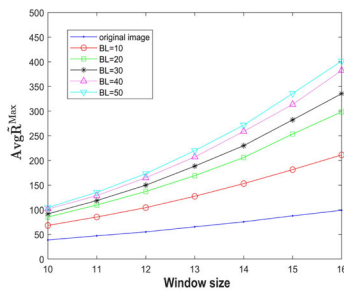
(d) Blur image at 30° blur angle



(e) Comparison of \bar{R}^{max} for 30° blur angle



(f) Blur image at 60° blur angle



(g) Comparison of \bar{R}^{max} for 60° blur angle

Fig. 7 (a) Test image: Mandrill-512×512. (b), (d) & (f) Blur images with 10°, 30° and 60° blur angle and 50 pixels blur length, respectively. (c), (e) & (g) Comparison of the average value of \bar{R}^{max} for various blur lengths- 10, 20, 30, 40, and 50 pixels. Here note that BL means blur length

4 Experimental results and discussion

In this section, we present the experimental results and evaluation of the performance of the proposed method.

4.1 Estimators, dataset and parameter setting

This subsection describes implementation of the proposed estimators on test images. First, we discuss the significance of the estimator $\tilde{R}^{max}(\mathbf{z})$ given in (10). To validate the effectiveness of (10), we consider a test image of size 512×512 pixels; Lena and Mandrill which are displayed in Figs. 6(a) and 7(a), respectively. To test \tilde{R}^{max} in the blurred and non-blurred images, we generated blurred images with different blur parameters. Here, we have used the linear blur model mentioned in (2) with different blur parameters, i.e., blur lengths ranging from 10, 20, ..., and 50 pixels, for each of the blur length we applied three blur angles (10° , 30° , and 60°). By picking various pairs of blur lengths and blur angles, we created blur images. Subsequently, considered 100,000 sub-images for window size ranging 10×10 to 30×30 from random locations and \tilde{R}^{max} is calculated for each sub-images. Moreover, the average value of \tilde{R}^{max} is compared between blurred and non-blurred images which are displayed in Figs. 6 and 7. Here non-blur image is considered as original image. The wide gap in \tilde{R}^{max} values of blurred and non-blurred images is observed from both the test images which are shown in Fig. 6(b-g) and Fig. 7(b-g). This phenomena are used effectively for segmenting blurred and non-blurred region. Moreover, it is seen that \tilde{R}^{max} of blurred image is always greater than of non-blurred image for a given window size, which can be seen from Figs. 6(b-g) and 7(b-g). The optimal window size should be above 10 pixels according to [18]. Therefore, we have chosen 10×10 to 16×16 pixels, window size for computation. For clarity, the comparison of \tilde{R}^{max} values from blur and nonblur images are calculated for two test images - Lena and Mandrill, which is shown in Figs. 6(c),(e) & (g) and 7(c),(e) & (g) for blur angles 10° , 30° & 60° , respectively. The function \tilde{R}^{max} is segregating blurred and non blurred region effectively and therefore using the value of \tilde{R}^{max} , we constructed local blur estimator. Next, we discuss $\hat{\theta}^l(\mathbf{z})$ and the proposed estimator $\tilde{B}(\mathbf{z})$ along with two weight functions $k_1(\mathbf{z})$ and $k_2(\mathbf{z})$. From Fig. 9(b), it is observed that $\hat{\theta}^l(\mathbf{z})$ produces uniform blur information in blurred region, whereas in a non-blurred region it gives nonuniform

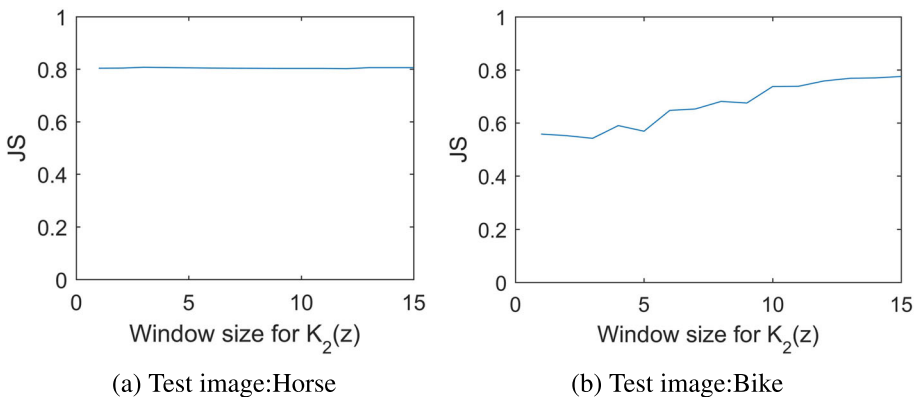


Fig. 8 : Jaccard Similarity (JS) index for $k_2(\mathbf{z})$ for various window size

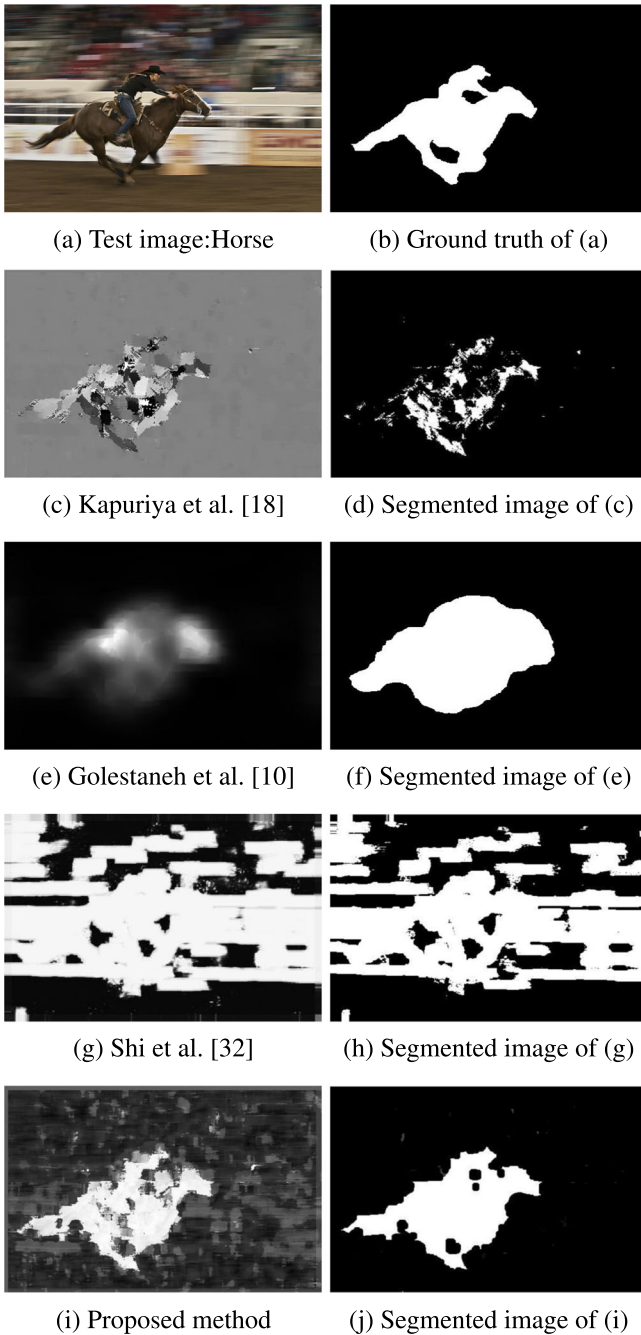


Fig. 9 (a) Input image: Horse-533×355; (b) Ground truth, (c) Blur information from [18], (d) Segmented object [18], (e-f) Blur information and segmented object using method from [10], (g-h) Blur information and segmented object using method from[32], (i-j) Blur information and segmented object using proposed method

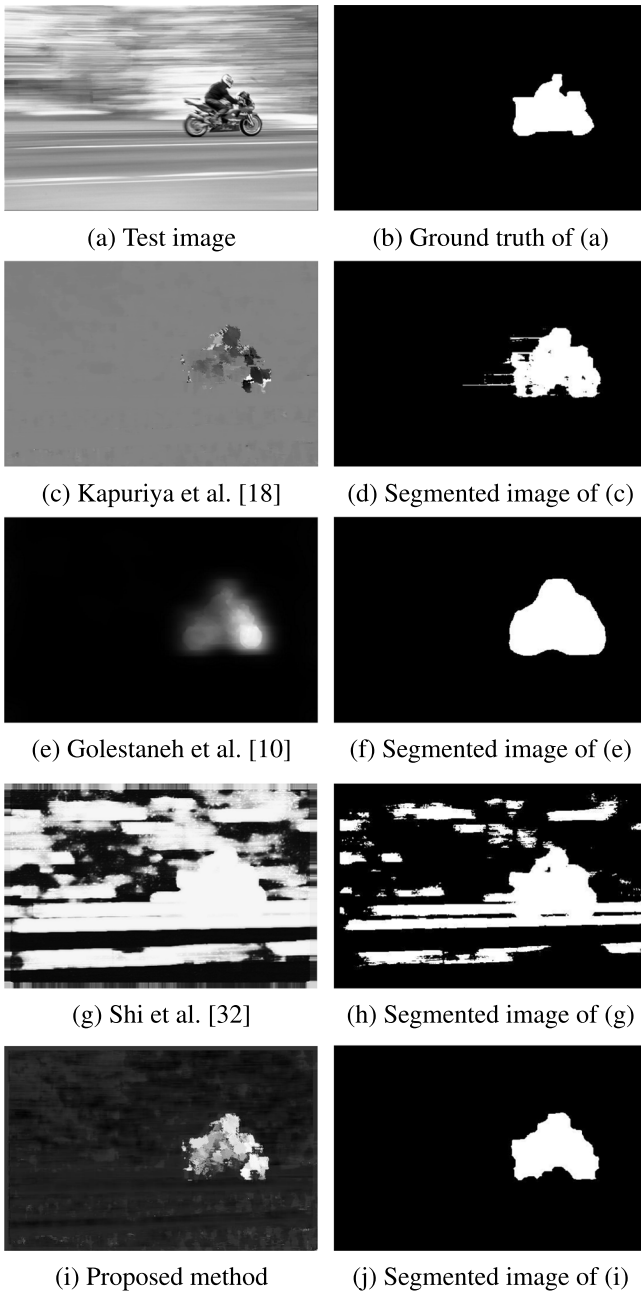


Fig. 10 (a) Input image: Bike- 533×355 ; (b) Ground truth, (c) Blur information and using method from [18], (d) Segmented object from [18], (e-f) Blur information and segmented object using method in [10], (g-h) Blur information and segmented object using method in [32], (i-j) Blur information and segmented object using proposed method

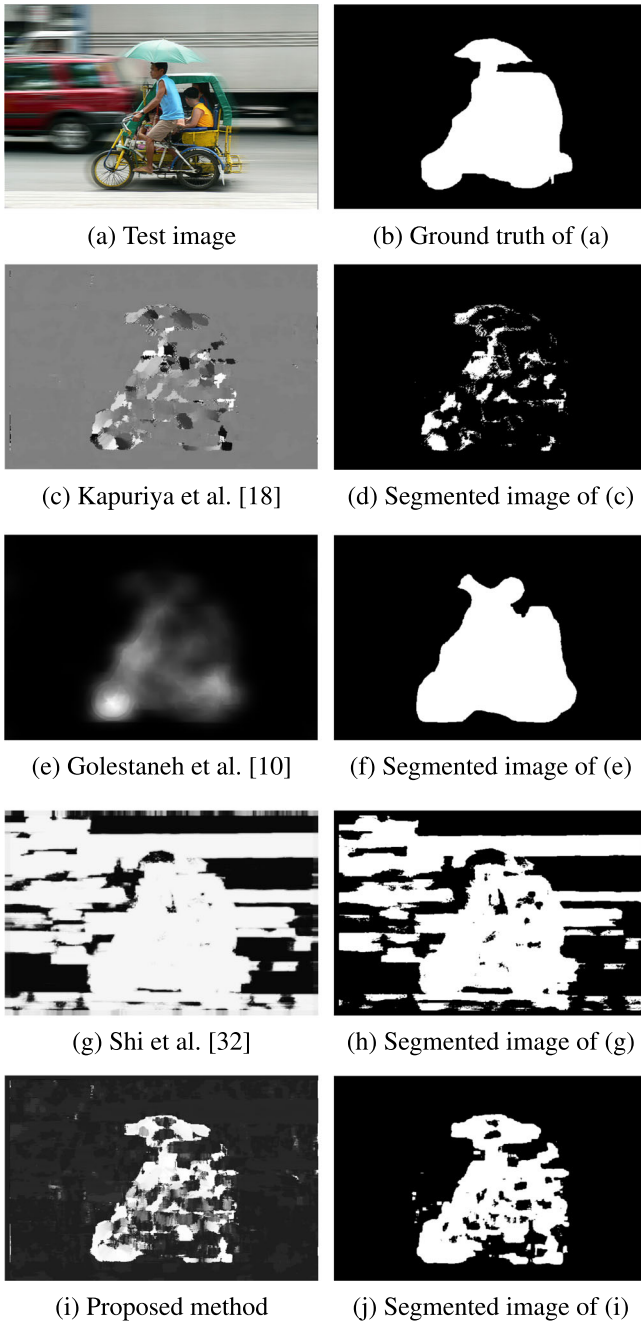
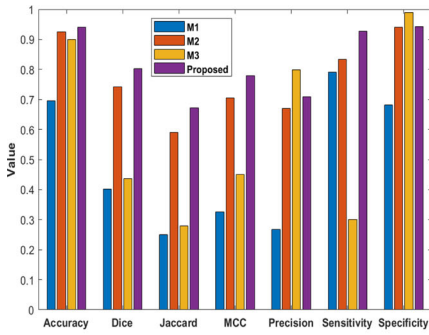
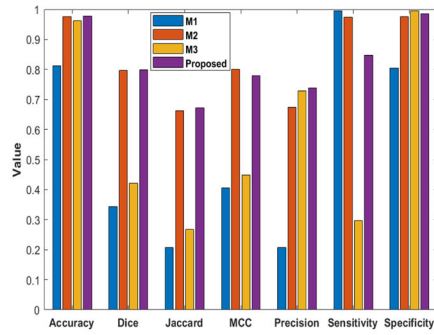


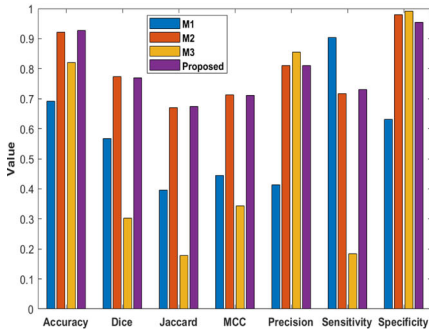
Fig. 11 (a) Input image: Rickshaw-640×450; (b) Ground truth, (c) Blur information and using method from [18], (d) Segmented object [18], (e-f) Blur information and segmented object using method in [10], resp. (g-h) Blur information and segmented object using [32] method, resp. (i-j) Blur information and segmented object using proposed method, resp



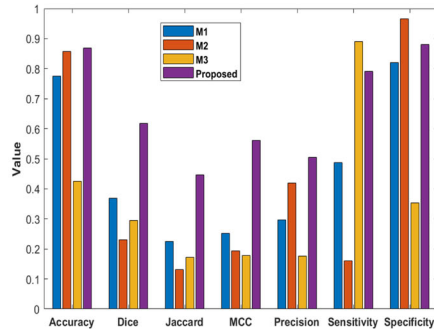
(a) Accuracy measures for the image in Fig. 9



(b) Accuracy measures for the image in Fig. 10



(c) Accuracy measures for the image in Fig. 11



(d) Accuracy measures for the image in Fig. 13

Fig. 12 Accuracy measures

blur information. The same can be observed in other test image displayed in Fig. 10(b). To overcome this problem, we used a new blur estimator $\tilde{B}(\mathbf{z})$ as mentioned in (8). This is tested on both grayscale and colour images taken from datasets of [32], <http://www.cse.cuhk.edu.hk/~leo/jia/projects/dblurdetect/dataset>. From Fig. 9(e), it is seen that the new blur estimator in the proposed method produces uniform output in the non-blurred regions, unlike the method proposed in [18]. The final segmentation result of proposed method is compared with [10, 18, 32] along with ground truth results which is seen in Fig. 9(f-j). By subjective observation, it can be seen that the outcomes from the proposed method is better than other existing methods. Subsequently, quantitative analysis has been done by computing various performance measures after segmentation. For segmentation, we have used Ostu’s method. The results are discussed in Section 4.2.

Further, we have analysed weight functions $k_2(\mathbf{z})$ and $k'_2(\mathbf{z})$ for different scenarios. Here, we observed that the geodesic distance based function is better in comparison to local variance based function. Fig. 8(a) & (b) shows Jaccard’s similarity index for $k_2(\mathbf{z})$ and $k'_2(\mathbf{z})$ derived over different window sizes ranging from 4×4 to 30×30 pixels. Here, we observed that any window size in the range greater than 4×4 for the function $k_2(\mathbf{z})$ produces a desirable result for the test image shown in Fig. 9(a). Whereas for second test image shown in Fig. 10, the window size greater than 10×10 produces a better result. This is due to the low contrast in the image. The third weight function $k''_2(\mathbf{z})$ mentioned in (13) is fully dependent on ϵ which can differ from image to image as mentioned previously.

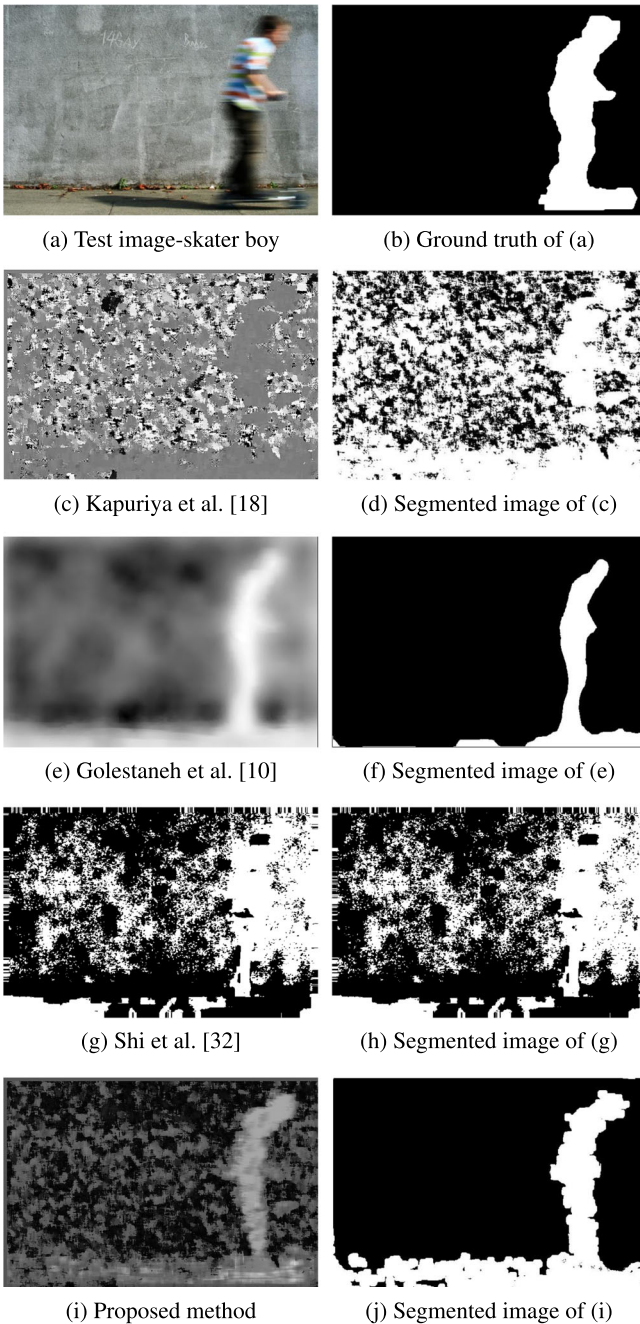


Fig. 13 (a) Input image: Boy-640×426;(b) Ground truth, (c) Blur information and using method from [18], (d) Segmented object [18], (e-f) Blur information and segmented object using method in [10], resp. (g-h) Blur information and segmented object using [32] method, resp. (i-j) Blur information and segmented object using proposed method, resp

Table 1 Average of all accuracy measures

Methods	Horse (Fig. 9)	Bike (Fig. 10)	Rickshaw (Fig. 11)	Skater boy (Fig. 13)
M1 [32]	0.4722	0.5145	0.5772	0.4604
M2 [10]	0.7692	0.8323	0.7932	0.4222
M3 [18]	0.5737	0.5672	0.4966	0.3555
Proposed	0.8224	0.8296	0.7935	0.6670

4.2 Quantitative performance

To assess the quantitative performance of our method, we compare our approach with M1, M2, and M3 methods described in [10, 18, 32], respectively. Here, M1 means method from [32], M2 means method from [10] and M3 means method from [18]. We determined various performance measures like; Dice index, Jaccard index, Accuracy, Sensitivity (Recall), Precision, Matthews Correlation Coefficient (MCC), and Specificity for three test images to find the similarity between the segmented image and the ground truth mask provided with the dataset. We have used a confusion matrix comprised of four values true positive (TP), true negative (TN), false positive (FP) and false-negative (FN), to determine these measures, and calculated as follows:

$$\text{Dice} = \frac{2TP}{(2TP + FP + FN)}, \quad \text{Jaccard} = \frac{TP}{(TP + FP + FN)},$$

$$\text{Accuracy} = \frac{TP + TN}{(TP + TN + FP + FN)},$$

$$\text{Sensitivity} = \frac{TP}{(TP + FN)}, \quad \text{Precision} = \frac{TP}{(TP + FP)}, \quad \text{Specificity} = \frac{TN}{(TN + FP)} \text{ and}$$

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}.$$

The values of these measures are closer to 1 indicates superior performance. The calculated values for three images are shown in Fig. 12. For the test image shown in Fig. 9, 5 out of 7 measures-accuracy, Jaccard, MCC, Dice, and sensitivity, are highest for the proposed method among all methods, which can be seen from Fig. 12(a). Values of other measures for the proposed method are also better. Though the M3 method gives the highest precision and specificity. For the test image shown in Fig. 10, 4 out of 7 measures- accuracy, precision, Jaccard, and Dice are highest for the proposed method, which can be seen from Fig. 12(b).

Table 2 Computation time (in seconds)

Methods	Horse (Fig. 9)	Bike (Fig. 10)	Rickshaw (Fig. 11)	Skater boy (Fig. 13)
M1 [32]	198	241	365	456
M2 [10]	100	113	181	275
M3 [18]	390	400	710	1085
Proposed	420	512	823	1095

Table 3 The value of parameters for proposed method

Methods	Horse (Fig. 9)	Bike (Fig. 10)	Rickshaw (Fig. 11)	Skater boy (Fig. 13)
α	1	50	10	0.1
β	100	100	100	100
ϵ_g	0.001	0.001	0.001	0.001
β_g	1000	1000	1000	1000

M2 techniques are likewise delivering the most noteworthy estimations of MCC. At the same time, the M1 method produces the highest sensitivity, and M3 produces the highest specificity. For the test image shown in Fig. 11, 3 out of 7 measures- accuracy, Jaccard, and MCC are highest for the proposed method, which can be seen from Fig. 12(a). Whereas Dice are highest in the M2 method, and sensitivity is highest from the M1 method. In contrast, precision and specificity are maximum in the M3 method. From the test image displayed in Fig. 13, 5 out of 7 measures- accuracy, Dice, Jaccard, MCC, and precision are highest for the proposed method, which can be seen from Fig. 12(d). Overall, the proposed method gives 80 % (in 17 out of 21) highest values across all measures in all test images. Averages performance of segmentation measures for all test images is presented in Table 1 shows the superior performance of the proposed method. The computational expense for all test images is recorded in Tables 2, and 3 and the proposed method takes extra computational time compared to other existing methods. The extra computational expense of the proposed method is because of LBAE, which is a time-consuming operation. Moreover, geodesic distance-based weight function $k_2(\mathbf{z})$ is also time taking operation. To reduce time complexity, other weight functions can be used, but it will deteriorate blur/nonblur segmentation accuracy. However, clearly visible that the performance of the proposed method is superior to other existing methods in terms of segmentation accuracy. We have used MATLAB software to implement the algorithm on a 2.0 GHz processor with 4GB RAM.

5 Conclusion

In this paper, a novel method to detect a partial blur region is proposed which facilitates object restoration, information retrieval, blur increment, etc., by segmenting blurred / non-blurred regions. In the proposed method, we have developed a new estimator with the help of the Radon transform and Laplacian of Gaussian operators. This estimator comprises of LBAE and \tilde{R}^{max} (see, (10)) along with two new weight functions (see, (11)). The experimental results carried out in this paper conclude the following: (i) As compared to other existing methods, the method proposed here is effective in detection & segmentation of the blurred region. (ii) Also, using various accuracy measures, the method proposed is better matched quantitatively to the ground truth result. (iii) The two-weight functions play an essential role in deciding the proportionality of index and maximum value of Radon transform. (iv) The proposed method is helpful in segregating blurred and non-blurred regions effectively, which further can be used as input for deblurring techniques. (v) It is more time-consuming than other existing methods, and we will attempt to find approaches to reduce the computational expense in the future.

Acknowledgements The authors would like to thank the Defence Institute of Advanced Technology, Pune, and Centre for Airborne Systems, Bangalore, for providing infrastructure for research work.

Funding This study received no specific grant from any funding agency in the public, commercial, or non profit sectors.

Data Availability The data set used in this paper is available with the corresponding author upon a reasonable request.

Declarations

Ethical Approval Not applicable and no human or animal subjects were involved.

Concent for Publication All the authors approved the final version of the paper.

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Chakrabarti, A, Zickler, T, Freeman WT (2010) Analyzing spatially-varying blur. In: IEEE conference on computer vision and pattern recognition, pp 2512–2519
2. Bahrami, K, Kot AC, Fan J (2013) A novel approach for partial blur detection and segmentation. In: IEEE international conference on multimedia and expo, pp 1–6
3. Bini AA, Bhat MS (2014) A nonlinear level set model for image deblurring and denoising. *Vis Comput* 30:311–325
4. Cai JF, Ji H, Liu C, Shen Z (2012) Framelet based blind motion deblurring from a single image. *IEEE Trans Image Process* 21(2):562–572
5. Cai JF, Osher S, Shen Z (2009) Linearized bregman iterations for frame-based image deblurring. *SIAM J Imaging Sci* 2(1):226–252
6. Chan TF, Wong CK (1998) Total variation blind deconvolution. *IEEE Trans Image Process* 7(3):370–375
7. Cho TS, Paris S, Horn BKP, Freeman WT (2011) Blur kernel estimation using the radon transform. In: IEEE conference on computer vision and pattern recognition, pp 241–248
8. Couzinie-Devy F, Sun J, Alahari K, Ponce J (2013) Learning to estimate and remove non-uniform image blur. In: IEEE conference on computer vision and pattern recognition, pp 1075–1082
9. Favaro P, Soatto S (2004) A variational approach to scene reconstruction and image segmentation from motion-blur cues. In: IEEE conference on computer vision and pattern recognition
10. Alirez Golestaneh S, Karam LJ (2017) Spatially varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In: IEEE conference on computer vision and pattern recognition, pp 5800–5809
11. Harmeling S, Michael H, Schoelkopf B (2010) Space-variant single-image blind deconvolution for removing camera shake. In: Proc. of neural information processing systems
12. Huang R, Feng W, Fan M, Wan L, Sun J (2018) Multiscale blur detection by learning discriminative deep features. *Neuro Comput* 285:154–166
13. Javaran TA, Hassanpour H, Abolghasemi V (2017) Automatic estimation and segmentation of partial blur in natural images. *Vis Comput* 33:151–161
14. Pan J, Hu Z, Su Z, Lee H-Y, Yang M-H (2015) Soft-segmentation guided object motion Deblurring. In: IEEE conference on computer vision and pattern recognition, pp 459–468
15. Ji H, Liu C (2008) Motion blur identification from image gradients. In: IEEE conference on computer vision and pattern recognition, pp 1–8
16. Ji H, Wang K (2012) A two-stage approach to blind spatially-varying motion deblurring. In: IEEE conference on computer vision and pattern recognition, pp 73–80
17. Kalalembang E, Usman K, Gunawan IP (2009) DCT-based local motion blur detection. In: International conference on instrumentation, communication, information technology, and biomedical engineering, pp v1–6
18. Kapuriya BR, Pradhan D, Sharma R (2019) Detection and restoration of multi-directional motion blurred objects. *Signal Image Video Process* 13(5):1001–1010
19. Kim TH, Lee KM (2014) Segmentation-free dynamic scene deblurring. In IEEE conference on computer vision and pattern recognition, pp 2766–2773
20. Kim B, Son H, Park S, Cho S, Lee S (2018) Defocus and motion blur detection with deep contextual features. *Pacific Graphics* 37(7):277–288

21. Krahmer F, Lin Y, McAdoo B, Ott K, Wang J, Widemann D et al (2006) Blind image deconvolution: Motion blur estimation. *Inst Math Appl Univ Minnesota Minneapolis Minnesota Tech Rep* 2133–5
22. Liew A (2006) Blind motion deblurring using image statistics. In: *Proc. of neural information processing systems*, pp 841–848
23. Liu R, Li Z, Jia J (2008) Image partial blur detection and classification. In: *IEEE conference on computer vision and pattern recognition*, pp 1–8
24. Liu S, Wang H, Wang J, Cho S, Pan C (2015) Automatic blur kernel size estimation for motion deblurring. *Vis Comput* 31:733–746
25. Ma K, Fu H, Liu T, Wang Z, Tao D (2018) Deep blur mapping?: Exploiting high level semantics by deep neural networks. *IEEE Trans Image Process* 27(10):5155–5166
26. Almeida Mariana S. C, Almeida Luis B (2010) Blind and semi-blind deblurring of natural images. *IEEE Trans Image Process* 19(1):36–52
27. Mariana SC, Almeida M, Figueiredo, Almeida M (2013) Parameter estimation for blind and nonblind deblurring using residual whiteness measures. *IEEE Trans Image Process* 22(7):2751–2763
28. Mignotte M (2005) An adaptive segmentation-based regularization term for image restoration. In: *IEEE international conference on image processing*
29. Oliveira JP, Figueiredo MAT, Bioucas-Dias JM (2014) Parametric blur estimation for blind restoration of natural images: linear motion and out-of-focus. *IEEE Trans Image Process* 23(1):466–477
30. Paramanand C, Rajagopalan AN (2013) Motion blur for motion segmentation. In: *IEEE international conference on image processing*, pp 4244–4248
31. Sakano M, Suetake N, Uchino E (2006) Robust identification of motion blur parameters by using angles of gradient vectors, In: *International symposium on intelligent signal processing and communications*, pp 522–525
32. Shi J, Xu L, Jia J (2014) Discriminative blur detection features, In: *IEEE conference on computer vision and pattern recognition*, pp 2965–2972, *Blur Detection Dataset*. <http://www.cse.cuhk.edu.hk/leo/jia/projects/dblurdetect>
33. Su B, Lu S, Tan CL (2011) Blurred image region detection and classification. In: *International conference on multimedia* pp 1397–1400
34. Sun H, Desvignes M, Yan Y, Liu W (2009) Motion blur parameters identification from radon transform image gradients, In: *Annual conference of IEEE industrial electronics*, pp 2098–2103
35. Tai YW, Tan P, Brown MS (2011) Richardson-lucy deblurring for scenes under a projective motion path. *IEEE Trans Pattern Anal Mach Intell* 33(8):1603–1618
36. Kim TH, Nah S, Lee KM (2016) Dynamic Scene deblurring using a locally adaptive linear blur model. *Computer Vision*
37. Tang C, Wu J, Hou Y, Wang P, Li W (2016) A spectral and spatial approach of coarse to fine blurred image region detection. *IEEE Signal Processing Letters* 23(11):1652–1656
38. Yan R, Shao L (2016) Blind image blur estimation via deep learning. *IEEE Trans Image Process* 25(4):1910–1921
39. Oyamada Y, Asai H, Saito H (2011) Blind deconvolution for a curved motion based on cepstral analysis. *IPSN Transactions on Computer Vision and Applications* pp 32–43
40. Zhang X, Burger M, Bresson X, Osher S (2010) Bregmanized non local regularization for deconvolution and sparse reconstruction. *SIAM J Imaging Sci* 3(3):253–276

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.