Front. Comput. Sci., 2022, 16(2): 162702
https://doi.org/10.1007/s11704-020-9526-x

RESEARCH ARTICLE

Defocus blur detection using novel local directional mean patterns (LDMP) and segmentation via KNN matting

Awais KHAN[1], Aun IRTAZA[2], Ali JAVED (✉)[3,4], Tahira NAZIR[1], Hafiz MALIK[2], Khalid Mahmood MALIK[4], Muhammad Ammar KHAN[1]

1 Department of Computer Science, University of Engineering and Technology, Taxila 47050, Pakistan
2 Department of Electrical and Computer Engineering, University of Michigan, Dearborn, MI 48128, USA
3 Department of Software Engineering, University of Engineering and Technology, Taxila 47050, Pakistan
4 Department of Computer Science and Engineering, Oakland University, MI 48309, USA

Abstract Detection and segmentation of defocus blur is a challenging task in digital imaging applications as the blurry images comprise of blur and sharp regions that wrap significant information and require effective methods for information extraction. Existing defocus blur detection and segmentation methods have several limitations i.e., discriminating sharp smooth and blurred smooth regions, low recognition rate in noisy images, and high computational cost without having any prior knowledge of images i.e., blur degree and camera configuration. Hence, there exists a dire need to develop an effective method for defocus blur detection, and segmentation robust to the above-mentioned limitations. This paper presents a novel features descriptor *local directional mean patterns* (LDMP) for defocus blur detection and employ KNN matting over the detected LDMP-Trimap for the robust segmentation of sharp and blur regions. We argue/hypothesize that most of the image fields located in blurry regions have significantly less specific local patterns than those in the sharp regions, therefore, proposed LDMP features descriptor should reliably detect the defocus blurred regions. The fusion of LDMP features with KNN matting provides superior performance in terms of obtaining high-quality segmented regions in the image. Additionally, the proposed LDMP features descriptor is robust to noise and successfully detects defocus blur in high-dense noisy images. Experimental results on Shi and Zhao datasets demonstrate the effectiveness of the proposed method in terms of defocus blur detection. Evaluation and comparative analysis signify that our method achieves superior segmentation performance and low computational cost of 15 seconds.

Keywords defocus blur detection, local directional mean patterns, image matting, sharpness metrics, blur segmentation

1 Introduction

Blur detection and segmentation is a challenging problem in image processing; as blur reduces the sharpness and contrast details of an image. However, a good quality image consists of clear and sharp objects. In modern digital photography, various focusing techniques are intentionally used to highlight the foreground objects and make the remaining part of the image blurry. When such images are encountered in object detection, the detection of sharp objects becomes challenging due to blurred image regions. Blur detection and segmentation are used in various applications that require the information wrap only in sharp regions, such as object detection [1], image matting [2], and image segmentation [3].

Blurring is usually caused by two factors, i.e., motion blur and defocus blur. Several factors, such as the movement of camera or objects, contribute to the motion blur of images. Whereas, defocus blur is a visual effect intentionally introduced by the photographers to highlight human subjects or foreground objects. Motion and defocus blur cause the loss of sharpness details in the image, and accurate object detection from these blurred regions becomes challenging. In this case, detection and segmentation of sharp and blurred regions become essential so that restoration algorithms such as video deblurring [4], sharp region detection [5], etc. can be applied to effectively restore the images.

Conventional approaches for the estimation of defocus blur depends on multiple images [6−8], where images of the different scenes are taken using similar focus settings. These approaches have certain limitations such as the requirement of a static scene, occlusion problem, etc. Nowadays, several approaches have been proposed to estimate the defocus blur from a single image without having any prior information about the blur level, type or focus settings [9−15]. Analysis of defocus blur with a single image can be categorized into frequency-based [5,13,16], depth-intensity based [17−20] and local-sharpness based approaches [7,21,22].

Frequency-based defocus blur detection method was proposed in [13] by generating the coherent blur map via evaluating the local frequency spectrum of the gradient fields. This method [13] only extracts the color edge and flat area infor-

Received December 27, 2019; accepted May 5, 2020

E-mail: ali.javed@uettaxila.edu.pk

mation to produce a coherent map. In [23], Xiao et al. proposed a multi-scale SVD fusion-based method that is used to perceive the defocus blur in the smooth region. To detect the defocus map, Chakrabarti et al. [24] presented the Point spread function (PSF) based on local frequency analysis. The drawback of this approach [24] is the incorrect labeling of regions in the final detected map. Su et al. [8] proposed a Hough transformation based framework for blur detection and classification. In [25] Golestaneh and Karam proposed "High frequency multi-scale fusion and sort transform" HiFST algorithm to segment the sharp and blur regions.

Depth based methods are also used for segmentation of defocus blur using information about the blur edges. In [26], Zhao et al. proposed a cross-ensemble network for diversity enhancement based on multiple smaller defocus detectors to differentiate sharp and blur regions. However, this approach is unable to estimate the parameters accurately and is computationally more complex. Liu et al. [19] proposed distinct local features i.e., power bands, saturation, association congruence, and gradient histogram to determine the blur type. Furthermore, blur measures such as the variance of wavelet coefficients [27], higher-order statistics and local variance [20] are also used for blur segmentation in a narrow depth of field images.

Most frequently used approaches for defocus blur segmentation are based on local sharpness measurement. However, these techniques need to specify a particular sharpness metric for an image. In [10], Zhu et al. proposed a method for computing point spread function (PSF) at the pixel level. However, computation is based on the local frequency component analysis [24] and only operates with defocus images. Shi et al. [6] introduced peculiar local sharpness features, i.e., kurtosis and gradient histogram span for defocus blur segmentation of local image regions. However, this method [6] is unable to detect defocus blur in homogeneous smooth regions in the image.

Recently, in [28] Shi et al. proposed a new sparse feature dictionary for blur detection based on an external set of defocus images. However, this method [28] is unable to perform well in the case of immense blurred regions in the image. In [29], Pang et al. proposed a kernel-specific feature descriptor for the detection of blur images. Additionally, they employed two different kernels for motion and defocus blur. Moreover, SVM was used for classification. In [30], Kim et al. proposed a deep learning method based on convolutional Neural Network (CNN) to detect the motion and defocus blur. Moreover, they employed the multi-scale reconstruction loss function with both low- and high-level spectral features to detect blur around the edges. In [31] Park et al. proposed a fully connected neural network that works on both the deep and handcrafted features. Convolutional neural network was employed to extract the deep features. In [32], Tang et al. proposed a deep neural network "DeFusionNET" that merges and refine deep features for the detection of defocus blur. In [21], the Local Binary Pattern (LBP) descriptor was employed for defocus blur detection. However, this method is unable to achieve better performance for images containing a high density of noise as LBP is sensitive to noise specifically in uniform or flat areas.

Existing state-of-the-arts have several limitations to detect image blurring i.e., discriminating sharp smooth and blurred smooth regions, low recognition accuracy in noisy images and high computational cost. Additionally, existing defocus blur detection methods are unable to accurately discriminate between the sharp smooth and blurred smooth regions. Local texture patterns descriptors achieve better accuracy and efficiency in various applications i.e., facial expression recognition [33], image retrieval [34], etc. However, the performance of existing local texture patterns degrades significantly for noisy images. To address the aforementioned challenges, we propose a novel features extraction method *Local Directional Mean Patterns* (LDMP) for defocus blur detection in images containing high-density of noise. This paper presents a novel method for blur detection and segmentation to retrieve the sharp and blur regions from defocus blurry images without quantifying the extent of blurriness. The main contributions of the proposed research work are as follows:

- For feature extraction, we propose a novel local directional mean patterns (LDMP) approach to determine the LDMP-Trimap from the image by utilizing the LDTP higher and lower patterns that are robust to noisy images.
- We employ an image matting based blur segmentation approach to segment the blur and sharp regions of the image from the proposed LDMP-Trimap.
- Rigorous experimentation was performed against several state-of-the-art methods over SHI and ZHAO datasets containing different distortions and high-dense noisy images to prove the effectiveness of the proposed framework.

The rest of this paper is organized as follows. Section 2 describes the proposed blur detection and segmentation model. Experimental results and analysis are presented in Section 3. Section 4 demonstrates the discussion and the conclusion is provided in Section 5.

2 Proposed method

The proposed method consists of two phases i.e., blur detection and blur segmentation. In the first phase, we perform blur detection to extract LDMP-Trimap using the local directional triplicate patterns (LDTP) and local directional mean patterns (LDMP). Initially, we used LDTP to compute higher and lower patterns of the blurry image. Later, we extract the LDTP to obtain the LDMP that provides the optimal sharp and blur patterns of the image. In order to obtain the LDMP-Trimap, we employ flood filling over the obtained patterns that display the detected sharp and blur regions.

In the second phase, we apply KNN matting over the LDMP-Trimap image to segment the sharp and blur regions. For KNN matting, LDMP-Trimap is convolved over the input image to produce the binarized image of the sharp and blur regions. More specifically, the resultant binary image represents the blur region as background denoted by white pixels and sharp regions represent the foreground objects depicted through black pixels as shown in Fig. 1.

2.1 Blur detection

In the first phase, we perform the defocus blur detection using

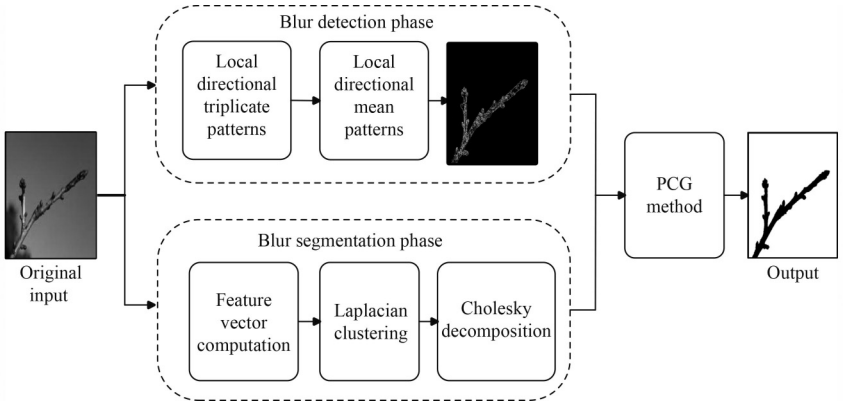

Fig. 1 Flow diagram of blur detection and segmentation

the LDTP and LDMP techniques for the extraction of LDMP-Trimap from the blurry image. We used the input image I_{db} consisting of sharp and blur regions to perform defocus blur detection as follows:

$$I_{\mathrm{db}} = c_r + b_r, \tag{1}$$

where c_r and b_r represents the clear and blur regions of the image. If I_{db} consists of RGB colors, it must be first transformed into a grayscale image I_{gs}.

$$I_{gs} = \mathrm{gry}(I_{\mathrm{db}}). \tag{2}$$

2.2 Local directional triplicate pattern (LDTP)

We present the local directional triplicate patterns (LDTP) that are computed by applying a pre-processing filter to Local Ternary patterns (LTP) to extract the higher and lower patterns of the image. In LDTP, firstly we employ the rank selection median filter on I_{gs} to reduce the visual distortion of the image as shown in Eq. (3).

$$I_{\mathrm{m}f} = f_{med}(I_{gs}), \tag{3}$$

where f_{med} represents the median filter and I_{mf} consists of filtered smooth patterns of the image. Now, we extract the local directional triplicate patterns $M(I_{mf(n)}, I_{mf(c)}, \Delta_t)$ from the image I_{mf} obtained by the rank selection median filtering. For this purpose, we used three-level thresholding along with a user-specified threshold on the image structure as follows:

$$M(I_{mf(n)}, I_{mf(c)}, \Delta_t) = \begin{cases} 1, & I_{mf(n)} \geqslant I_{mf(c)} + \Delta_t, \\ 0, & I_{mf(n)} > I_{mf(c)} - \Delta_t \& I_{mf(n)} < I_{mf(c)} + \Delta_t, \\ -1, & I_{mf(n)} \leqslant I_{mf(c)} - \Delta_t, \end{cases} \tag{4}$$

where n is the neighbouring pixels and c represents the center value of the matrix. The threshold Δ_t used in Eq. (4) controls the sharpness sensitivity of the LDTP method. More specifically, if Δ_t is set to a higher value (i.e., 0.2, 0.3, etc.) then it retrieves the least number of higher and lower patterns of the image and if Δ_t is set to a lower value (i.e., 0.001) then it extracts more patterns that consist of sharp and blur pixels. The impact of the LDTP threshold Δ_t is shown in Fig. 2. In the LDTP computation, we define the optimal threshold Δ_t alongwith the central pixel value instead of setting the threshold using only the central pixel value. For example, if the central pixel has an integral value 35 and $\Delta_t = 10$ then the ranges from $\mathrm{I}_{\mathrm{m}f(c)} - \Delta_t$ to $\mathrm{I}_{\mathrm{m}f(c)} + \Delta_t$ results in 25 to 45 respectively. The integral value "0" is assigned to the pixel within the interval of $\mathrm{I}_{\mathrm{m}f(c)} - \Delta_t$ to $\mathrm{I}_{\mathrm{m}f(c)} + \Delta_t$, whereas, assigns the value of "1" to the pixel if it is greater than $\mathrm{I}_{\mathrm{m}f(c)} + \Delta_t$ and "–1" if lesser than $\mathrm{I}_{\mathrm{m}f(c)} - \Delta_t$. For efficient defocus blur detection, higher and lower patterns of the image must be computed separately. Whereas, LDTP use triplicate condition to extract higher and lower patterns of the image collectively as computed in Eq. (4). Therefore, LDTP features $M(I_{mf(n)}, I_{mf(c)}, \Delta_t)$ are further divided into higher and lower patterns of the image represented as Local Directional Higher Patterns (LDHP) and Local Directional Lower Patterns (LDLP). However, to compute LDHP and LDLP, we transform the negative values from $M(I_{mf(n)}, I_{mf(c)}, \Delta_t)$ into positive values as shown in Fig. 3.

To obtain the LDHP features all the "–1" values from $M(I_{mf(n)}, I_{mf(c)}, \Delta_t)$ are converted into '0" while leaving the rest of the values unchanged as shown in Eq. (5). Consequently, the resultant $LDHP$ comprises of all positive higher patterns of the image.

$$LDHP = \left\{ M(I_{mf(n)}, I_{mf(c)}, \Delta_t) = -1 \rightarrow 0 \right\}, \tag{5}$$

Similarly, the LDLP features are obtained by replacing the value "1" into "0" and "–1" into "1" from $M(I_{mf(n)}, I_{mf(c)}, \Delta_t)$

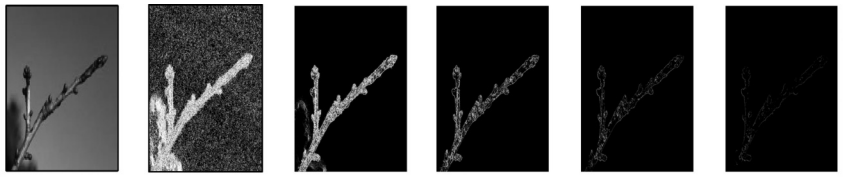

Fig. 2 Impact of LDTP Threshold on the input image to preserve sharpness map

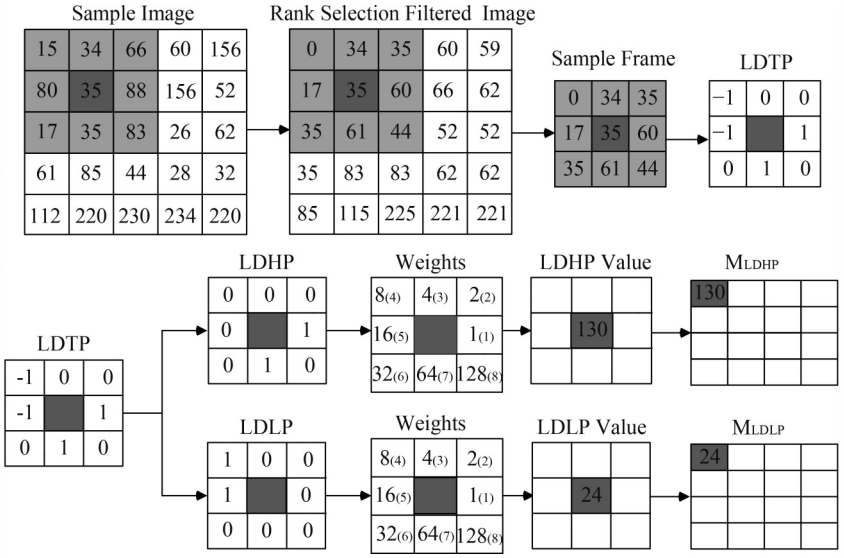

Fig. 3 Local directional triplicate pattern computation

as shown in Eqs. (6) and (7). Finally, the resultant $LDLP$ features contain all positive lower patterns of the image.

$$LDLP = \left\{M(I_{mf(n)}, I_{mf(c)}, \Delta_t) = 1 \to 0\right\}, \tag{6}$$

$$LDLP = \left\{M(I_{mf(n)}, I_{mf(c)}, \Delta_t) = -1 \to 1\right\}. \tag{7}$$

$LDHP$ and $LDLP$ values represent the binary bit streams that must be transformed into decimal values. For this purpose, we start extraction of bits from the east followed by moving in a counter-clockwise direction to convert the bit streams into equivalent decimal numbers as follows:

$$M_{LDHP} = \sum_{i=8}^{1} LDHP(C_{\mathrm{rn}}) \times 2^{i-1}, \tag{8}$$

$$M_{LDLP} = \sum_{i=8}^{1} LDLP(C_{rn}) \times 2^{i-1}, \tag{9}$$

where C_{rn} is the right neighbor pixel of the center pixel in a matrix M_{LDHP} and M_{LDLP}. Finally, we obtain the M_{LDHP} and M_{LDLP} representing the higher and lower patterns of the image separately. Consequently, both patterns contain the specific sharpness details of the input image that can be used efficiently for further LDMP processing (i.e., M_{LDHP} contain more detailed left side sharp patterns and M_{LDLP} contains more specific right side sharp patterns of the image). Shown in Fig. 4 is an example of the patterns obtained by higher LDMP (M_{LDHP}) and lower LDMP (M_{LDLP}).

2.2.1 Local directional mean patterns (LDMP)

After obtaining the LDTP, we extract two separate higher M_{LDHP} and lower M_{LDLP} patterns from the input image. Now we need to extract optimal Local Directional Mean Patterns ($LDMP$) from M_{LDHP} and M_{LDLP} patterns. For this purpose, we need to verify the M_{LDHP} and M_{LDLP} patterns for the co-occurrence of sharp pixels. We perform this task in a two-step process. First, we compute the mean vector of both M_{LDHP} and M_{LDLP} patterns separately as follows:

$$MV_{LDHP} = \frac{\sum_{i=1}^{n} M_{LDHP}}{n}, \tag{10}$$

$$MV_{LDLP} = \frac{\sum_{i=1}^{n} M_{LDLP}}{n}. \tag{11}$$

After computing the mean vectors in Eqs. (8) and (9), we calculate the mean value from both calculated mean vectors as follows:

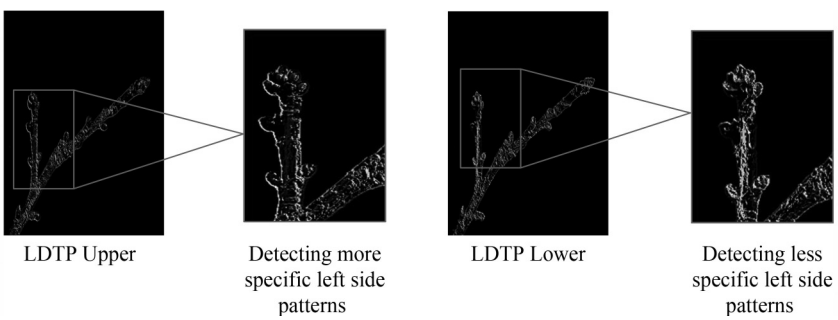

Fig. 4 Patterns comparison of LTP Upper and LTP Lower

$$LDHP_{(mean)} = \frac{\sum_{i=1}^{n} MV_{LDHP}}{n}, \quad (12)$$

$$LDLP_{(mean)} = \frac{\sum_{i=1}^{n} MV_{LDLP}}{n}. \quad (13)$$

By computing the mean value from the mean vectors, we authenticate the co-occurrence of sharp pixels from both M_{LDHP} and M_{LDLP} patterns and merged them into a single optimal $LDMP$. We retain the values higher than their mean values from M_{LDHP} and M_{LDLP} patterns and concatenated them together to obtain the optimal sharp features $LDMP$ as shown in Eq. (12). Additionally, we set the values less than their mean values to "0" as shown in Fig. 5.

$$LDMP = [M_{LDHP} > LDHP_{(mean)} \cup M_{LDLP} > LDLP_{(mean)}]. \quad (14)$$

There exists a connectivity gap between sharp pixels in $LDMP$ that need to be filled to generate the LDMP-Trimap. For this purpose, we employ the flood filling method to fill these gaps. Flood filling method is applied on the $LDMP$ to process the connectivity gap surrounded by filled pixels with the specified color as shown in Fig. 6 and represented as follows:

$$TM_{LDMP} = ff(LDMP), \quad (15)$$

where ff represents the flood filling method and TM_{LDMP} denotes the LDMP-Trimap obtained from LDMP as shown in Fig. 7.

2.3 KNN matting

In the second phase, we apply KNN matting over the detected LDMP-Trimap from $LDMP$. We employ the KNN matting to perform matching over the blur and non-blur neighborhood pixels using multiple layers. KNN matting implements the non-local concept of alpha matting that decomposes the Trimap image into the background and foreground layers represented as "bg" and "fg" respectively. We used the default threshold of KNN matting for foreground and background layers of the input image as:

$$\begin{aligned} fg &= TM_{LDMP}, \quad \text{if} \quad (TM_{LDMP} > 0.99), \\ bg &= TM_{LDMP}, \quad \text{if} \quad (TM_{LDMP} < 0.01), \end{aligned} \quad (16)$$

$$Im = fg + bg, \quad (17)$$

where TM_{LDMP} represents the LDMP-Trimap image. KNN method produces the segmented result in four phases i.e., γ-Feature Computation, Laplacian clustering, Cholesky decomposition, and PCG method.

2.3.1 γ- feature computation

In KNN matting, γ-features computation includes the extraction of spatial coherence details of the input color image having center pixels "k" surrounded by the neighboring pixels nn. The γ-feature computation consists of two parts. The first part provides the colors and image magnitude details and the second part provides the spatial factor information computed as:

$$\begin{aligned} \gamma_F &= RS_F(Im, r\times c, d_{ig})\prime; [a;b]/Z + T, \\ Z &= \sqrt{(r\times r + c\times c)}\times L, \\ T &= rand(2, r\times c)\times 1e^{-6}, \end{aligned} \quad (18)$$

where RS_F, Img, and L represents the reshape function, input image, and the level respectively. [a,b] is computed from the function $I2S_F$ that determines the equivalent subscript values corresponding to a given single index into an array as:

$$[a, b] = I2S_F([r\ c], 1 : r\times c). \quad (19)$$

We used γ_F to compute the row-wise diagonal values of foreground layer "fg_{dig}" and background layer "bg_{dig}" as follows:

$$R_D = \gamma_F(fg_{dig}: bg_{dig})/100, \quad (20)$$

where R_D contains the sparse sharp and blur patterns of the image. However, we need to extract the maximum sharp and blur pixels information. For this purpose, we extract the maximum sharp and blur values M_{val} by applying the absolute function on the patterns R_D as follows:

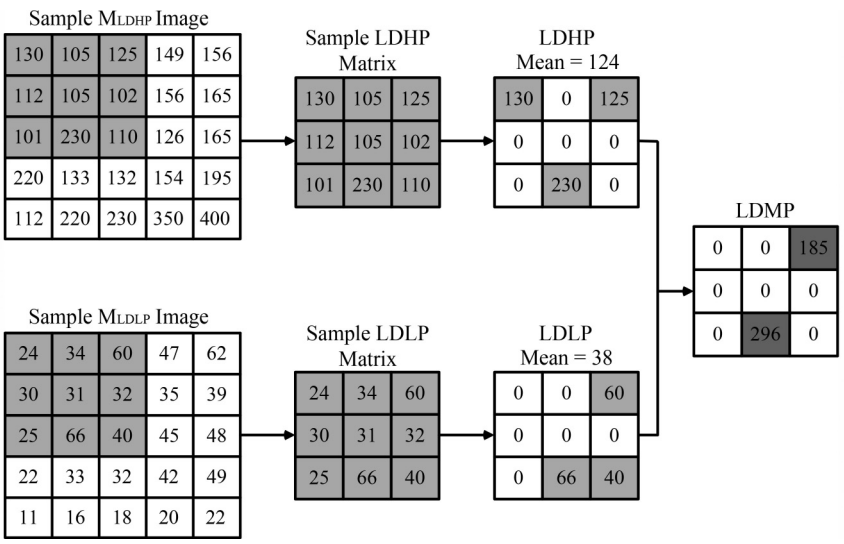

Fig. 5 Local directional mean pattern computation

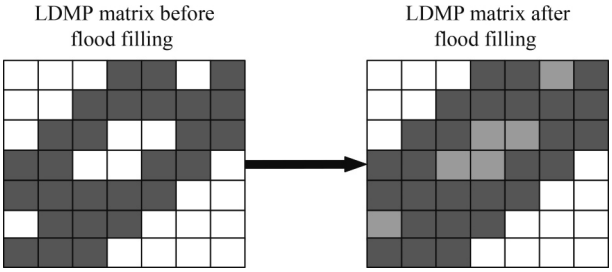

Fig. 6 Flood filling computation

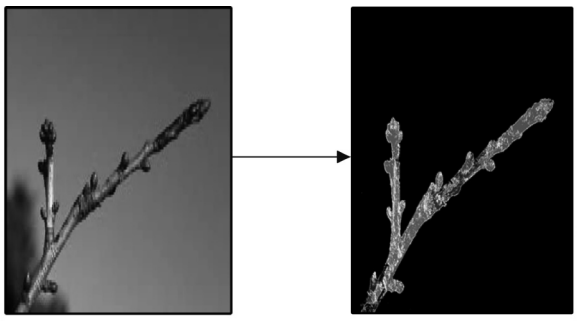

Fig. 7 LDMP-trimap output

$$M_{val} = \max(1 - \sum_{i=1}^{n} (abs(R_D, row(:,1) - R_D, row(:,2)))/(d_{ig} + 2), 0). \quad (21)$$

2.3.2 Laplacian clustering

After computing the M_{val} we employ the clustering to combine the maximum nearest neighbor's pixels into similar clusters. In KNN matting, Laplacian clusters generate comparatively better results over Gaussian clusters in terms of negotiable edge losses. For Laplacian clustering LC first, we need to extract the R_{val} from the M_{val} as:

$$R_{val} = S_r(L(:,1), L(:,2), M_{val}, r \times c, r \times c), \quad (22)$$

where L contains the row values for each layer having 1 for foreground and 2 for the background layer and $r \times c$ represents the dimension of R_{val}. Next, we compute the sparse patterns that contain the diagonal values of R_{val} as follows:

$$D_{val} = S_d\left(\sum (R_{val}, 2), 0, r * c, r * c\right). \quad (23)$$

Finally, we apply the Laplacian clustering LC_i containing the set of multiple clusters as $LC_i = \{LC_1, LC_2, ..., LC_i\}$ on R_{val} and D_{val} from M_{val} to create the clusters of neighboring values as:

$$LC_i = D_{val} - R_{val}. \quad (24)$$

As LC_i includes multiple clusters of nearest regions so, we have to combine all the clusters collectively to generate the sparse patterns H denoting the sharp and blur regions. For this purpose, we convolve and combine the LDMP-Trimap image TM_{LDMP} with D_{val} and LC_i to compute the sparse patterns H as follows:

$$H = LC_i + \lambda * D_{val}(Im,\ 0,\ r \times c,\ r \times c), \quad (25)$$

where λ is a constant and Im represents the foreground and background layers of the Trimap image.

2.3.3 Cholesky Decomposition

The values obtained in the sparse patterns H are partially filled and there exists a connectivity gap between the values highlighting the blur and sharp regions. Therefore, this gap needs to be filled to obtain optimal segmented results. To address this issue, we employ "Cholesky Decomposition" CD over H with zero-fill to completely fill the patterns that can be expressed as:

$$H_{CD} = CD(H). \quad (26)$$

2.3.4 PCG method

The conjugate gradient method is used to solve unconstrained optimization problems i.e., energy minimization. Additionally, PCG increases the robustness of proposed blur detection and segmentation methods. Therefore, for H_{CD} optimization, we normalize the H_{CD} by convolving the lambda constant with foreground inside the Preconditioned Conjugate Gradients (PCG) along-with the complement of H_{CD} that returns the binarized segmented output X as follows:

$$X = PCG(H, \lambda * fg, H_{CD}, H_{CD}'). \quad (27)$$

Next, we apply the quantization process on the resultant X to minimize the effects of the illumination change in the image. If there exist higher illumination changes in an image, a higher quantization (i.e., 10.0, 12.5) is used to overcome that change. At last, a bilateral filter is applied over X to segment the blur and preserve the sharp regions of an image. KNN segmented results are shown in Fig. 8.

In addition, we also tested alpha matting for the segmentation over LDMP trimaps. However, the alpha matting failed to segment the minor edges of the sharp objects correctly. Consequently, the false positive rate of alpha matting is high in comparison to the KNN-matting. A comparison of KNN matting and alpha matting is shown in Fig. 9.

3 Experiments and results

3.1 Datasets

Performance of the proposed method is evaluated on Shi et al. [28] and Zhao et al. [35] datasets. The Shi dataset [28] consists of 1000 images where 296 images are of motion blur and the rest 704 are defocused. We used 704 out-of-focus images for defocus blur detection and segmentation. This dataset provides the hand-drawn ground truth images that indicate the blur and non-blur regions.

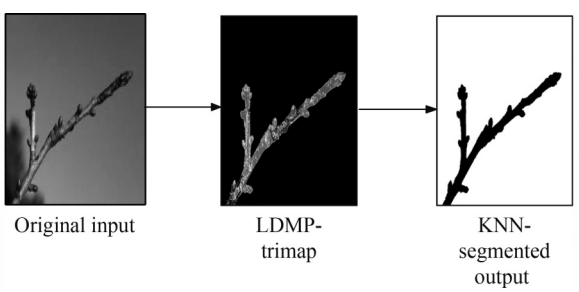

Fig. 8 KNN segmented results

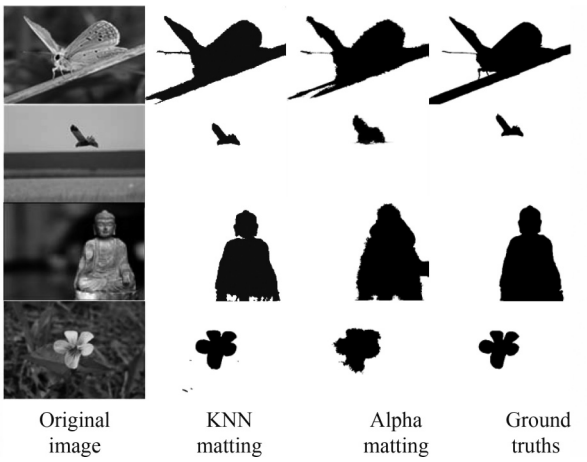

Fig. 9 Comparison of KNN matting and alpha matting

On the other hand, the Zhao dataset [35] consists of 500 images with pixel-wise annotated ground truth images. This dataset contains several low contrast, homogenous regions and cluttered background images that make this dataset more challenging for blur detection and segmentation.

3.2 Performance evaluation of proposed method

We evaluated the performance of the proposed method using precision, recall and f1-score metrics. We used these metrics for performance evaluation as also adopted by the comparative approaches. We computed the precision and recall as follows:

$$\mathrm{Pr} = \frac{T_P}{T_P + F_P}, \tag{28}$$

$$\mathrm{Re} = \frac{T_P}{T_P + F_N}, \tag{29}$$

where F_N represent the incorrectly estimated sharp regions, T_P denotes the correctly estimated blur regions and F_P represents the pixels incorrectly denoting blur regions available in the ground truth. In addition, we also computed the statistical measure F1-score that represents the harmonic mean of precision and recall. F1-score is a true indicator for performance comparison when some methods have better precision and others have better recall. F1-score lies in the range of 0 to 1, where 0 represents the lowest and 1 represents the highest value. We computed the F1-score as follows:

$$f1 = 2 \times \frac{\mathrm{Pr} \times Re}{\mathrm{Pr} + \mathrm{Re}}, \tag{30}$$

where Pr and Re represents the precision and recall of the proposed method. Shown in Table 1 are the values of precision, recall, and F1-score of the proposed blur detection and blur segmentation methods.

3.3 Comparative study

We also compared the performance of the proposed method with existing state-of-the-art techniques [6,8,19,21,24,25,28,29,31,32,35] along with LTP. More specifically, we provided a comparative analysis in terms of both qualitative and quantitative evaluation.

In qualitative evaluation, a visual comparison of Zhao and Shi datasets are shown in Figs. 10 and 11. As the results of comparative approaches are presented in binarized form, therefore, we also transform our LDMP-Trimap into binary for a fair comparison. Our proposed detection and segmentation techniques outperform all state-of-the-art techniques over both datasets regardless of camera configuration and scene interpretation. We evaluated the performance of the proposed technique against eleven existing state-of-the-art techniques [6,8,19,21,24,25,28,29,31,32,35] along with LTP on Shi and Zhao datasets. In [24], Chakrabarti et al. performed

Table 1 Performance evaluation of the proposed methods

		Precision	Recall	F1-score
SHI Dataset	Proposed blur detection	0.875	0.942	0.907
	Proposed blur segmentation	0.881	0.944	0.912
DUT Dataset	Proposed blur detection	0.910	0.867	0.88
	Proposed blur segmentation	0.907	0.898	0.89

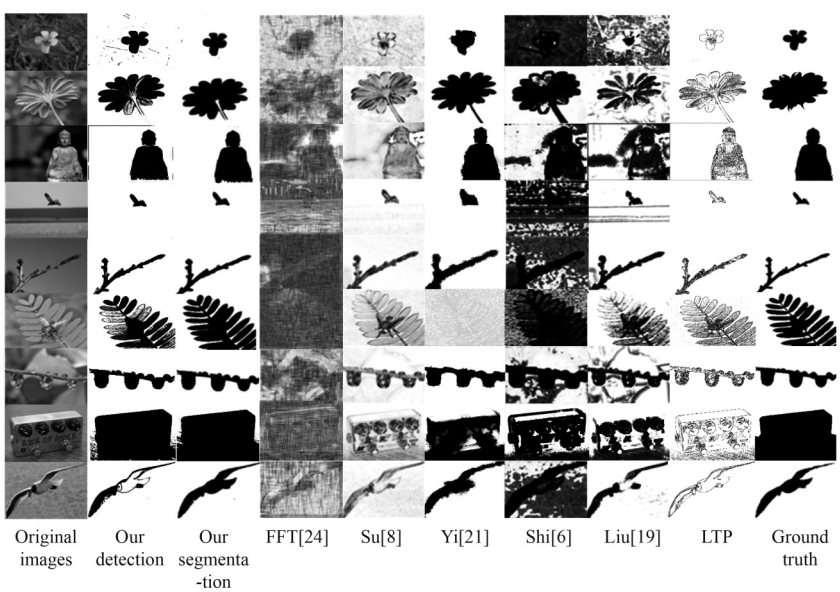

Fig. 10 Results achieved by different blur detection and segmentation methods on Shi dataset

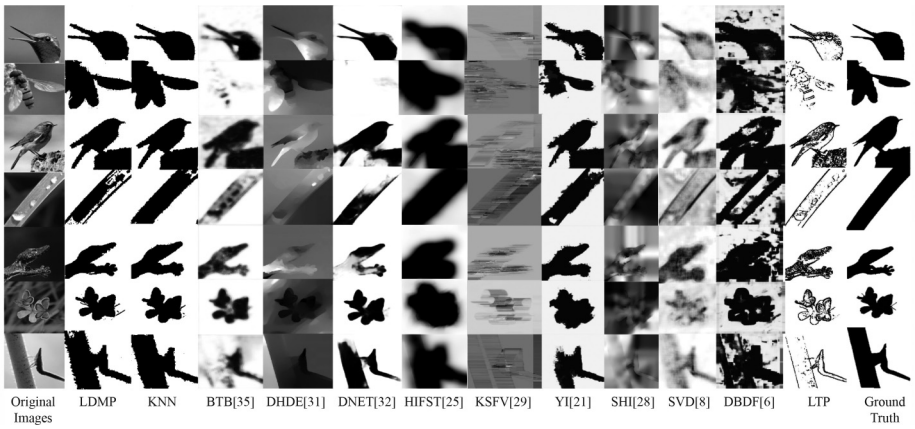

Fig. 11 Results achieved by different blur detection and segmentation methods on Zhao dataset

blur detection using the local Fourier transform by evaluating small neighborhood being blurred using the point spread function(PSF). However, the performance of this technique [24] degrades when performing blur detection under real-time conditions (i.e., camera configuration, scene interpretation, etc.). Liu et al. [19] performed the partial image blur detection and segment the image based on different types of blur i.e., defocus blur and motion blur. However, this technique [19] is unable to accurately segment the blur region due to ineffective edge detection. In [8], Su et al. performed defocus blur segmentation using the Hough transformation. Although, this technique [8] performs marginally better than [19,24] by preserving the shape of detected objects. However, this method [8] is unable to clearly segment the blur and non-blur regions. In [29] Pang et al. used kernel-specific features descriptor to train the SVM for blur detection, however, the method outperforms the Discriminative Blur Detection Features (DBDF) [6] method but still unable to achieve better accuracy. Shi et al. [6] employed various methods i.e. kurtosis, average power spectrum, and data-driven local filters comprising of Gabor and Laplacian to distinguish the blur and sharp regions in the input images. These methods [6] provide better blur detection performance for defocus images, however, the performance of these methods degrade significantly in high-dense noisy images. Shi et al. [28] used a sparse feature dictionary for blur detection based on an external set of defocus images. However, this method [28] is unable to perform well in the case of immense blurred regions in the image. Existing local binary patterns-based approaches i.e., [21] are unable to perform better segmentation on defocus images when encountered with noise. In [25], high-frequency DCT coefficients were used to detect the blur and non-blur regions, however unable to detect minor edges in the homogenous regions.

Few recent techniques [26,30−32,35] used deep learning frameworks i.e., convolutional neural networks (CNN) for blur detection in defocused images. However, these techniques have more computational cost as compared to approaches using conventional machine learning classifiers.

For quantitative evaluation, we provided the PR curves of the proposed and each comparative method on Shi and Zhao datasets in Figs. 12 and 13. More specifically, we used precision, recall, and F1-score metrics for performance comparison on Zhao and Shi datasets as shown in Fig. 14 and Fig. 15. Moreover, we presented separate curves and F1-score comparison for blur detection LDMP and segmentation KNN matting methods. The proposed KNN segmentation method achieves the highest precision, recall, and F1-score over comparative approaches. Additionally, the proposed blur detection LDMP

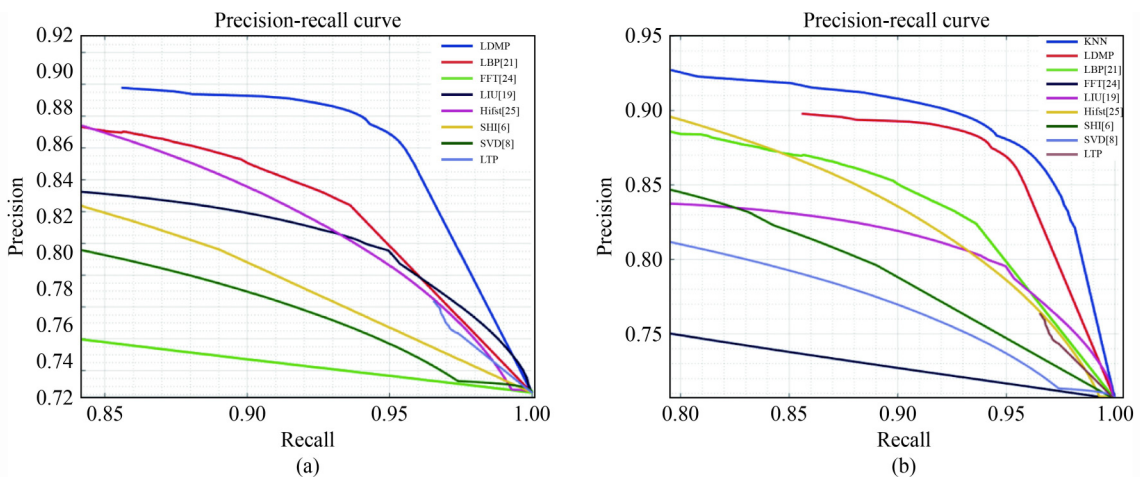

Fig. 12 Precision-Recall Curves for detection and segmentation on Shi-dataset: (a) Precision recall curve of blur detection via LDMP; (b) precision-recall curve of blur segmentation via KNN matting

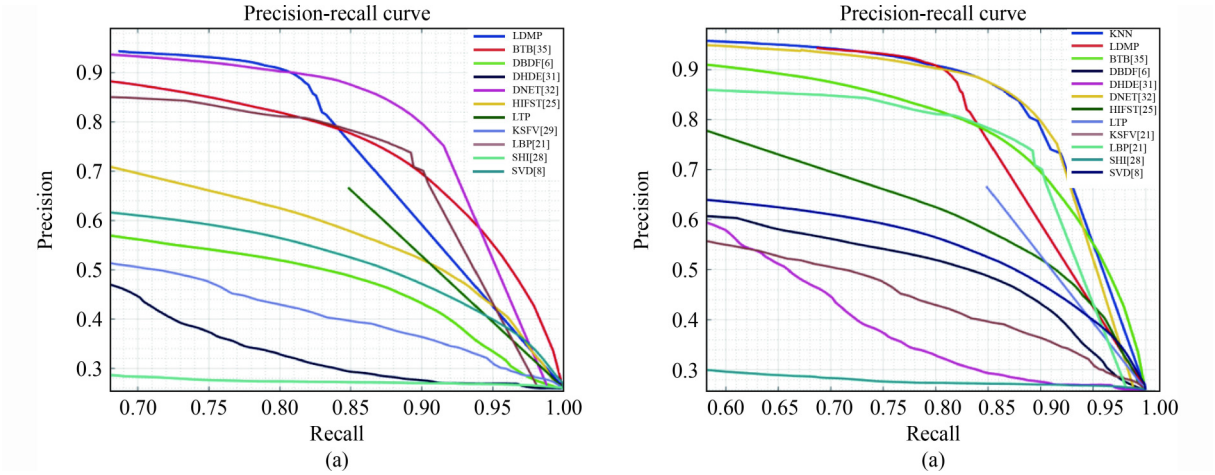

Fig. 13 Precision-Recall Curves for detection and segmentation on Zhao-dataset: (a) Precision recall curve of blur detection via LDMP; (b) precision-recall curve of blur segmentation via KNN matting

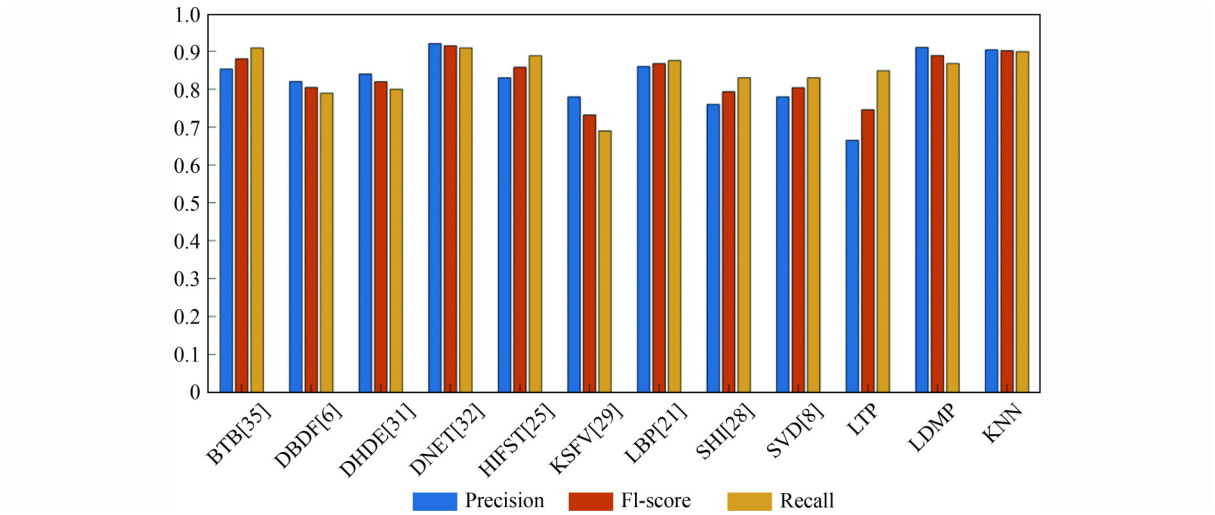

Fig. 14 F1-score comparison of proposed method with all comparative methods on Zhao-dataset

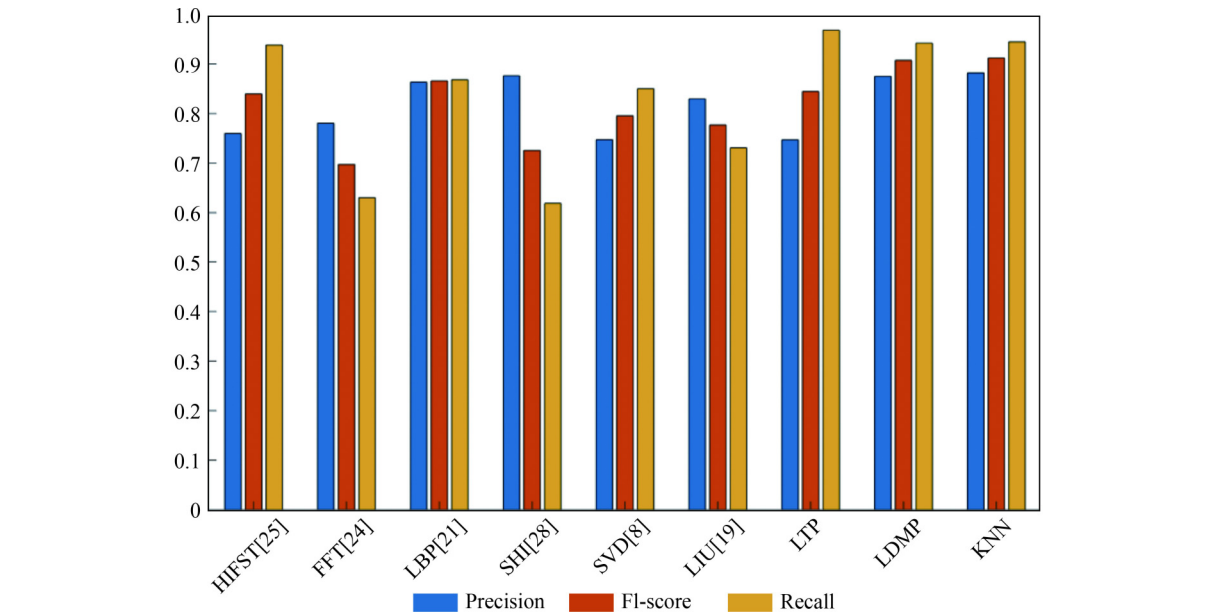

Fig. 15 F1-score comparison of proposed method with all comparative methods on Shi-dataset

method also provides superior detection performance on Shi [28] and Zhao [35] datasets over existing state-of-the-art methods. More specifically, the proposed blur detection method LDMP achieves the precision, recall and F1-score of 0.875, 0.942, and 0.907 on Shi dataset and 0.910, 0.867, and 0.88 respectively on Zhao-dataset. Similarly, our KNN matting-based blur segmentation method achieves the precision, recall and F1-score of 0.881, 0.944, and 0.912 on Shi-dataset and 0.907, 0.898, and 0.89 respectively on Zhao-dataset.

3.4 Time complexity analysis

We designed an experiment to compute the time complexity of the proposed and existing methods. We provided the time complexity values of the proposed and existing state-of-the-art [6−10,19,21,28,35] in Table 2 and Table 3. From the results, we can clearly observe that the proposed blur detection and segmentation methods are the second-best in terms of efficiency after LBP [21] as compared to existing systems. More precisely, we achieve the time complexity of 10 seconds, whereas [21] achieves 3.55 seconds and [35] has the largest computational complexity of 5 days. The main reason for the efficient implementation of the proposed blur detection and segmentation methods is the direct use of integral image values which makes the complexity independent of the size of the local region. For run-time computation, we implemented the code of comparative methods provided by the authors, as well as used our own implementations where code is not provided. Ten randomly selected images with an approximate size of 640×480 pixels are tested for time complexity analysis. The optimal threshold Δ_t is located within a certain range [0.0250−0.0990] in LDTP computation over the input image. Furthermore, we tuned the following parameters for KNN matting i.e., lambda = 100, level = 1 and nearest neighbors lies in the range of 3 (for material matting) to 15 (for natural image matting).

The values are selected after the detailed experimentation and the proposed methods produced the best results on these parameter settings. The Matlab 2018@ is used for the implementation of the proposed methods. Moreover, the proposed techniques are implemented on Intel(R) Core (TM) m3-7Y30 CPU @ 1.00 GHz, 1.61 GHz with an 8 GB memory system.

Table 2 Time complexity analysis of different blur detection methods

Sharpness metric	Avg. runtime
Gradient histogram span (mGHS) [6, 19]	273.19 s
Local Binary pattern (mlbp) [21]	3.55 s
Total variation (mTV) [7]	50.s00 s
Singular value decomposition (mSVD) [8]	38.66 s
Average power spectrum slope (mAPS) [6]	22.89 s
Proposed LDMP (Shi-Dataset)	15.00 s
Proposed LDMP (Zhao-Dataset)	10 s

Table 3 Time complexity analysis of different segmentation methods

Blur segmentation	Avg. runtime
Proposed KNN (Zhao-Dataset)	10 s
Proposed KNN (Shi-Dataset)	15 s
Shi [6]	705.27 s
LBP [21]	40 ms
Tang [32]	11.6 h
Zhao [35]	5 days
Su [8]	37 s
Shi [28]	38.36 s
Vu [7]	19.18 s
Zhuo and Sim [9]	20.59 s
Zhu [10]	12 min

4 Discussion

In this section, we compared the Local Binary Patterns (LBP) [21] approach with our proposed method. It has been observed that the LBP segmentation does not perform well when distinct depth discontinuity occurs between the background and foreground regions in the images. In [21], the sharpness of an image is estimated through local neighborhood processing. However, it is difficult to integrate areas with different sharpness extents within a local window, especially around the edges where the depth discontinuity occurs. Although the noise reduction method was employed in [21] for image denoising, however, LBP descriptor is still unable to provide better results as compared to the proposed method as shown in Fig.16. We can clearly observe from Fig.16 that the LBP descriptor is unable to effectively segment the image corrupted by the noise. To overcome this problem, we employed the proposed three-level thresholding to generate the LDTP and an optimal LDMP-Trimap. Where LDMP provides better performance as compared to the LBP descriptor as it is robust to noise and variation in illumination conditions. Blur detection and segmentation methods achieve better results than LBP in both cases, with noise removal and without noise removal as shown in Fig. 16. From the results, we can clearly observe that our proposed LDMP descriptor outperforms the LBP descriptor [21] for defocus blur detection and segmentation for high-dense noisy images. Although, our blur detection and segmentation approaches are computationally more expensive than LBP slightly. However, the algorithmic complexity of both approaches is *O N(logN)*. Hence from the results obtained after this experiment, we argue that our proposed LDMP is more effective and efficient descriptor over LBP for high-dense noisy images. It is also important to mention that the proposed framework was specifically designed for the detection of the sharp objects amongst the defocus objects. However, still able to detect the foreground objects even in the presence of motion blur as shown in Fig. 17. To evaluate the performance of the proposed framework on motion blur images, we also conducted some experiments on motion blur images from the Shi et al. [28] dataset. The proposed framework achieves better results on defocus blur images over motion blur images. We would like to mention that the performance of the proposed method slightly drops while discriminating sharp and smooth blur regions due to the consideration of a smaller local neighborhood. This limitation can be addressed by decreasing the threshold Δ_t value (i.e., 0.0120, 0.0100) as shown in Fig. 18. However, we plan to extend the proposed framework in terms of blur detection and segmentation for smooth blur and sharp blur regions along with other types of blur as well.

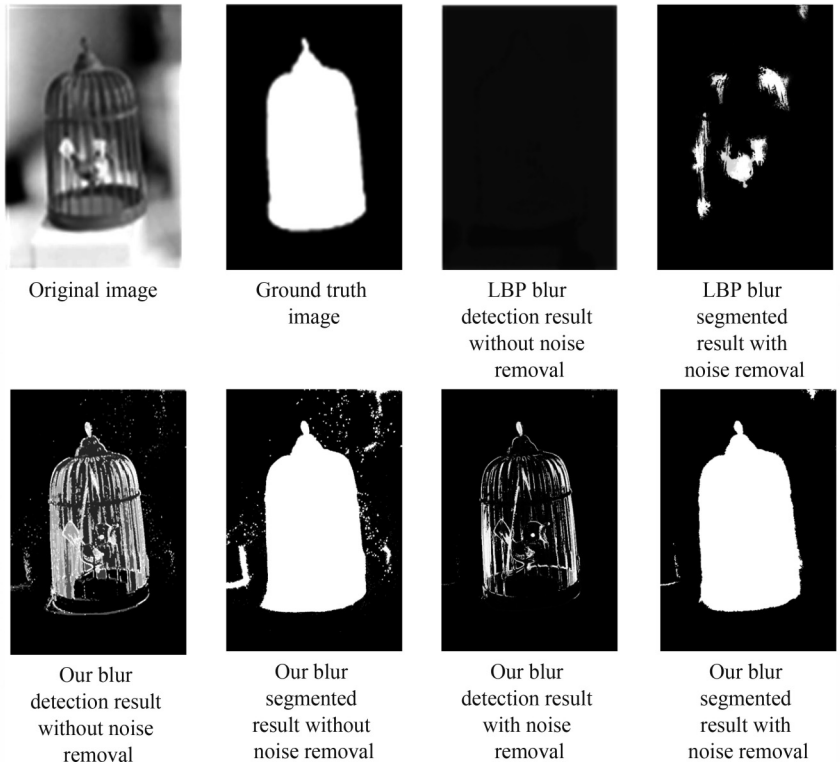

Fig. 16 LBP vs. LDMP comparison

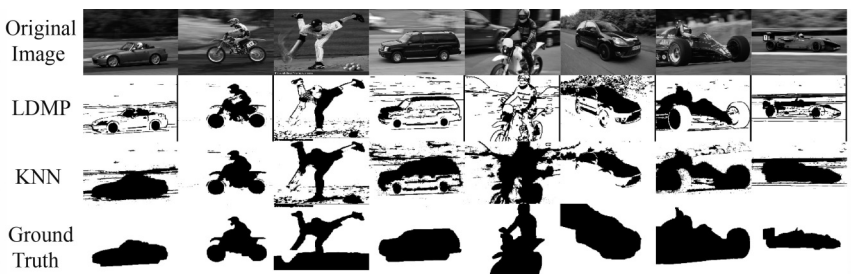

Fig. 17 LDMP and KNN results on motion blur

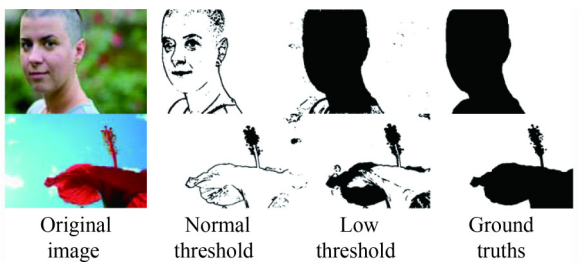

Fig. 18 Limitations of the proposed system

5 Conclusions and future work

This paper has addressed, defocus blur detection and segmentation problem without exploiting any information regarding camera configuration, level, and intensity of the blur. For this purpose, we present Local Directional Triplicate Patterns (LDTP) and novel Local Directional Mean patterns (LDMP) descriptors for the detection of blur and sharp regions in the image. Additionally, the proposed method extracts the LDMP-Trimap images using the local sharpness patterns of an input image for blur segmentation. In order to perform binarized segmentation of blur and non-blur regions, KNN matting is applied over the detected LDMP-Trimap. Moreover, the resultant binarized segmented output indicates the blur and sharp regions separately. Experimental results on Shi and Zhao datasets have demonstrated that the proposed method is robust to noise and is able to generate more accurate defocus maps over comparative methods. The proposed method provides superior detection performance as compared to existing systems. Additionally, our method is most efficient among all comparative methods that make it suitable for real-time processing. In the future, we plan to extend the proposed work by making it robust to homogenous regions and motion blur.

Acknowledgements This work was supported and funded by the Directorate ASR&TD of UET-Taxila.

References

1. Krishnamurthy B, Sarkar M. Deep-learning network architecture for object detection. U.S. Patents 10, 019, 655, 2018
2. Price B L, Schiller S, Cohen S, Xu N. Image matting using deep learning. Ed: Google Patents, 2019
3. Liu C, Liu W, Xing W. A weighted edge-based level set method based

on multi-local statistical information for noisy image segmentation. Journal of Visual Communication and Image Representation, 2019, 59: 89–107
4. Gast J, Roth S. Deep video deblurring: the devil is in the details. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 2019
5. Gvozden G, Grgic S, Grgic M. Blind image sharpness assessment based on local contrast map statistics. Journal of Visual Communication and Image Representation, 2018, 50: 145–158
6. Shi J, Xu L, Jia J. Discriminative blur detection features. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. 2014, 2965–2972
7. Vu C T, Phan T D, Chandler D M. S_3: a spectral and spatial measure of local perceived sharpness in natural images. IEEE Transactions on Image Processing, 2011, 21(3): 934–945
8. Su B, Lu S, Tan C L. Blurred image region detection and classification. In: Proceedings of the 19th ACM International Conference on Multimedia, Scottsdale, Arizona. 2011
9. Zhuo S, Sim T. Defocus map estimation from a single image. Pattern Recognition, 2011, 44(9): 1852–1858
10. Zhu X, Cohen S, Schiller S, Milanfar P. Estimating spatially varying defocus blur from a single image. IEEE Transactions on Image Processing, 2013, 22(12): 4879–4891
11. Tang C, Hou C, Song Z. Defocus map estimation from a single image via spectrum contrast. Optics letters, 2013, 38(10): 1706–1708
12. Zhang X, Wang R, Jiang X, Wang W, Gao W. Spatially variant defocus blur map estimation and deblurring from a single image. Journal of Visual Communication and Image Representation, 2016, 35: 257–264
13. Wing T Y, Brown M S. Single image defocus map estimation using local contrast prior. In: Proceedings of the 16th IEEE International Conference on Image Processing. 2009, 1797–1800
14. Shan Q, Jia J, Agarwala A. High-quality motion deblurring from a single image. ACM Transactions on Graphics (Tog), 2008, 27(3): 1–10
15. Rajabzadeh T, Vahedian A, Pourreza H. Static object depth estimation using defocus blur levels features. In: Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing. 2010, 1–4
16. Mavridaki E, Mezaris V. No-reference blur assessment in natural images using Fourier transform and spatial pyramids. In: Proceedings of IEEE International Conference on Image Processing (ICIP). 2014, 566–570
17. Lin J, Ji X, Xu W, Dai Q. Absolute depth estimation from a single defocused image. IEEE Transactions on Image Processing, 2013, 21(11): 4545–4550
18. Zhou C, Lin S, Nayar S K. Coded aperture pairs for depth from defocus and defocus deblurring. International Journal of Computer Vision, 2011, 93(1): 53–72
19. Liu R, Li Z, Jia J. Image partial blur detection and classification. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. 2008, 1–8
20. Tang C, Wu J, Hou Y, Wang P, Li W. A spectral and spatial approach of coarse-to-fine blurred image region detection. IEEE Signal Processing Letters, 2016, 23(11): 1652–1656
21. Yi X, Eramian M. LBP-Based Segmentation of Defocus Blur. IEEE Transactions on Image Processing, 2016, 25(4): 1626–1638
22. Hassen R, Wang Z, Salama M M. Image sharpness assessment based on local phase coherence. IEEE Transactions on Image Processing, 2013, 22(7): 2798–2810
23. Xiao H, Lu W, Li R, Zhong N, Yeung Y, Chen J. Defocus blur detection based on multiscale SVD fusion in gradient domain. Journal of Visual Communication and Image Representation, 2019, 59: 52–61
24. Chakrabarti A, Zickler T, Freeman W T. Analyzing spatially-varying blur. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2010
25. Golestaneh S A, Karam L J. Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, 5800–5809
26. Zhao W, Zheng B, Lin Q, Lu H. Enhancing diversity of defocus blur detectors via cross-ensemble network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, 8905–8913
27. Zhang Y, Hirakawa K. Blur processing using double discrete wavelet transform. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013, 1091–1098
28. Shi J, Xu L, Jia J. Just noticeable defocus blur detection and estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, 657–665
29. Pang Y, Zhu H, Li X, Li X. Classifying discriminative features for blur detection. IEEE Transactions on Cybernetics, 2015, 46(10): 2220–2227
30. Kim B, Son H, Park S J, Cho S, Lee S. Defocus and Motion Blur Detection with Deep Contextual Features. In: Proceedings of Computer Graphics Forum. 2018, 277−288
31. Park J, Tai Y W, Cho D, Kweon I S. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, 1736–1745
32. Tang C, Zhu X, Liu X, Wang L, Zomaya A. DeFusionNET: defocus blur detection via recurrently fusing and refining multi-scale deep features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019, 2700–2709
33. Nigam S, Singh R, Misra A. Local binary patterns based facial expression recognition for efficient smart applications. In: Hassanien A, Elhoseny M, Ahmed S, Singh A, eds. Security in Smart Cities: Models, Applications and Challenges. Springer, Cham, 2019, 297−322
34. Kumar G S, Mohan P K. Local mean differential excitation pattern for content based image retrieval. SN Applied Sciences, 2019, 1(1): 1–10
35. Zhao W, Zhao F, Wang D, Lu H. Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network. In: Proceedings of the IEEE Conference on Computer vision and Pattern Recognition. 2018, 3080–3088

Awais Khan is currently working toward the MS degree program as a full time research scholar at Department of Computer Science in University of Engineering and Technology, Pakistan. He graduated from University of Wah, Pakistan in 2017 with a Bachelor of Science degree in Computer Science. His research interests lie in computer vision, neural networks, machine learning and data science.

Aun Irtaza has completed his PhD in 2016 from FAST-nu, Islamabad Pakistan. During his PhD he remained working as a research scientist in the Gwangju Institute of Science and Technology (GIST), South Korea. He became an Associate Professor in 2017 and department of computer science chair in 2018 in the University of Engineering and Technology (UET) Taxila, Pakistan. He is currently working as visiting Associate Professor in the University of Michigan-Dearborn. His research areas include computer vision, multimedia forensics and big data analytics. He has more than 40 publications in IEEE, Springer, and Elsevier Journals.

Ali Javed received the BSc degree with honors and 3rd position in Software Engineering from UET Taxila, Pakistan in 2007. He received his MS and PhD degrees in Computer Engineering from UET Taxila, Pakistan in 2010 and 2016. He received Chancellor's Gold Medal in MS Computer Engineering degree. He is serving as an Assistant Professor in Software Engineering Department at UET Taxila, Pakistan. He has served as a Postdoctoral Scholar in SMILES lab at Oakland University, USA in 2019 and as a visiting PhD scholar in ISSF Lab at University of Michigan, USA in 2015. His areas of interest are image processing, computer vision, medical image processing, video content analysis, machine learning and multimedia forensics.

Tahira Nazir is currently working toward the PhD degree at Department of Computer Science in University of Engineering and Technology, Pakistan. She has done the MS(CS) from Department of Computer Science, UET Taxila, Pakistan in 2016. Her research interests are computer vision, medical imaging, machine learning and data science.

Hafiz Malik is an Associate Processor in the Electrical and Computer Engineering (ECE) Department at University of Michigan – Dearborn, USA. His research in the areas of automotive cybersecurity, IoT security, sensor security, multimedia forensics, steganography/steganalysis, information hiding, pattern recognition, and information fusion is funded by the National Science Foundation, National Academies, Ford Motor Company, and other agencies. He has published more than 100 papers in leading journals, conferences, and workshops. He is a founding member of the Cybersecurity Center for Research, Education, and Outreach at UM-Dearborn and member leadership circle for the Dearborn Artificial Intelligence Research Center at UM-Dearborn. He is also a member of the Scientific and Industrial Advisory Board (SIAB) of the National Center of Cyber Security Pakistan.

Khalid Mahmood Malik (Senior Member, IEEE) is currently an Assistant Professor with the School of Engineering and Computer Science, Oakland University, USA. His research interests include multimedia forensics, development of intelligent decision support systems using analysis of medical imaging and clinical text, secure multicast protocols for intelligent transportation systems, and automated ontology and knowledge graph generation. His research is supported by the National Science Foundation (NSF), Brain Aneurysm Foundation, and Oakland University.

Muhammad Ammar Khan recently, completed his MS degree in Computer Science from University of Engineering and Technology, Pakistan. He done his graduation in Computer Sciences from University of Wah, Pakistan in 2017. His research interests are computer visions, machine learning, data science and neural networks.