Defocus map estimation from a single image using improved likelihood feature and edge-based basis

Shaojun Liu[a,b,c], Qingmin Liao[a,b,c], Jing-Hao Xue[d], Fei Zhou[e,*]

[a]*Shenzhen Key Laboratory of Information Science and Technology, Shenzhen, 518055, China*
[b]*Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China*
[c]*Graduate School at Shenzhen, Tsinghua University, Shenzhen, 518055, China*
[d]*Department of Statistical Science, University College London, London, WC1E 6BT, UK*
[e]*College of Information Engineering, Shenzhen University, Shenzhen, 518060, China*

Abstract

Defocus map estimation (DME) is very useful in many computer vision applications and has drawn much attention in recent years. Edge-based DME methods can generate sharp defocus discontinuities but usually suffer from textures of the input image. Region-based methods are free of textures but cannot catch the defocus discontinuities very well. In this paper, we propose a DME method combining edge-based and region-based methods together to keep their respective advantages while eliminating the shortcomings. The combination is achieved via regression tree fields (RTF). In an RTF, the input feature and the linear basis are of vital importance. For our RTF, they are obtained as follows. (i) Two orthogonal gradient operators with the corresponding subsets of Gabor filters are employed in localized 2D frequency analysis to generate accurate likelihood, and the first K highest local maximums of likelihood are sent to an RTF as input feature. (ii) At the same time, the input image is processed by three edge-based methods and the results serve as the linear basis of RTF. The experiments demonstrate that the proposed method outperforms state-of-the-art DME methods. Moreover, the proposed method can be readily applied to defocused image deblurring and defocus blur detection.

Keywords: Defocus map estimation, Regression tree fields, Localized 2D frequency analysis

1. Introduction

Defocus blur is very common in images of three-dimensional scenes, especially when the aperture of the camera or the depth range of the scene is large. The defocus amount is often spatially-varying as the depth of the scene usually varies spatially [1]. Therefore, the defocus amount can be used as a cue of depth estimation [2]. Moreover, it can also apply to many other computer vision tasks, such as defocus magnification [3], image quality assessment [4], image focus editing [5], defocused image deblurring [6], saliency detection [7],

[*] Corresponding author
Email address: flying.zhou@163.com (Fei Zhou)

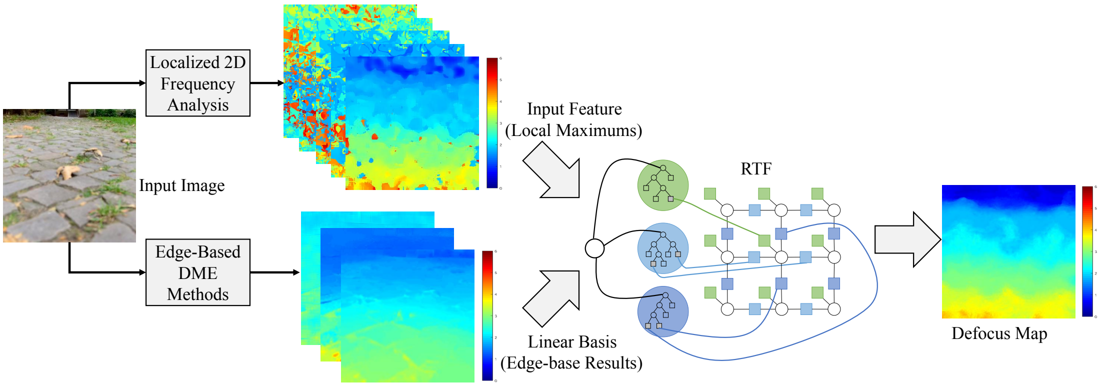

Localized 2D Frequency Analysis
Input Image
Edge-Based DME Methods
Input Feature (Local Maximums)
Linear Basis (Edge-base Results)
RTF
Defocus Map

Figure 1: Framework and an example of the proposed method.

multi-focus image fusion [8], foreground/background segmentation [9], and extension of depth of field [10]. Consequently, defocus map estimation (DME), a technique to estimate the defocus blur amount at each pixel of the input RGB image, is important and has attracted much attention in the past decades.

Early DME methods usually require at least two images captured with different camera settings [11–13]. This implicitly assumes the scene to be static to avoid parallax. Early single image DME methods are usually based on some specially-designed hardware, such as light field camera [14], coded aperture [15] and active illumination [16]. Though they can get good estimations, they employ some specially-designed hardware, which induces inconvenience for the majority of people. These shortcomings limit their applications.

In this paper, the goal is to develop a method for DME from a single image captured with conventional cameras. The framework and an example of the proposed method can be found in Figure 1. The first K highest local maximums of the likelihood function $L(r)$ with respect to defocus amount r are first obtained through localized 2D frequency analysis and then sent to a regression tree fields (RTF) [17] as input feature. At the same time, the input image is processed by three edge-based methods and the results serve as linear basis of the RTF. Finally, the final defocus map is predicted via RTF.

The motivations and contributions of this paper are summarized as follows. (i) In existing localized 2D frequency analysis [18], a gradient operator and a set of Gabor filters with all possible orientations are employed. However, if the angle between the orientations of the gradient operator and the Gabor filter is near to 90°, the filtering response will approximate to 0. Consequently, such Gabor filters can hardly provide any useful information. Therefore, to extract information as effectively as possible and obtain more accurate likelihood, two orthogonal gradient operators and the corresponding subsets of Gabor filters are employed for localized 2D frequency analysis in this paper. (ii) Based on the observation that the first several highest local maximums of likelihood can catch the true defocus amount with a high probability, in this paper, the first K highest local maximums of likelihood are extracted as the input feature of RTF. This feature can catch and present all the essential information with a low dimension. (iii) The extracted feature is region-based feature and usually cannot catch defocus discontinuities very well. Fortunately, edge-based methods can catch the discontinuities well. Therefore, in this paper, three typical edge-based methods are integrated into the framework by using them as the linear basis of RTF. Consequently, the proposed method is not only free from the influence of textures but also can catch defocus discontinuities reasonably well.

The rest of the paper is organized as follows. Section 2 gives a brief review of the related work. Section 3 describes the proposed method in detail. The experiments and discussions are shown in Section 4. Section 5 demonstrates that the proposed method can be readily applied to defocused image deblurring and defocus blur detection. Finally the paper is concluded in Section 6.

2. Related work

According to the way of analyzing defocus information, existing methods for DME from a single image captured with conventional cameras can be sorted into two categories, i.e., *edge-based* methods and *region-based* methods.

2.1. Edge-based methods

Defocused edges contain the information of defocus amount and there are many edges in natural images. Therefore, it is a natural cue to estimate defocus amount from edges. In [3], a two-step framework for DME is proposed: firstly, the defocus amount is estimated at edge pixels to get a sparse defocus map; then, the full defocus map is obtained via an interpolation guided by the input image. Though the defocus map might suffer from textures in the input images, this is the first approach to obtaining full defocus map and the framework has been widely used in other edge-based methods since then.

In [5], the position of edge points is analyzed at the sub-pixel level and the full defocus map is interpolated from the sparse defocus map at edge points in the same way as [3]. It can get better estimations at edge points than [3], however, it also suffers from the influence of textures. The authors of [19] propose to use local contrast to measure defocus amount at edge points and employ a graph-cut algorithm to get the full defocus map, which usually cannot catch the defocus discontinuities very well. In [20], the defocus amount is estimated by analyzing the ratio of the variances of the first- and second- order gradients. The algorithm is very fast, however, it can only get correct estimations at edge regions.

In [21], the ratio of the gradients of original and reblurred images at edge points is analyzed to obtain a sparse defocus map, and the full defocus map is interpolated from the sparse map by using Laplacian matting [22] with the input image as guidance. This is a very typical edge-based method and many other latter methods borrow its idea. The authors of [23] prove that the interpolation can be viewed as an α-matting procedure, and use guided image filter [24] to speed up the interpolation. In [25], spectrum contrast is calculated to measure defocus amount and the relationship between them is established to transform spectrum contrast to defocus amount. The authors of [26] adaptively select the strength of the reblurred kernel used in [21]. In [27], non-isolated edges are preprocessed before reblurring to eliminate their undesirable influence. The authors of [28] take the local image patch near an edge point as a matrix and use its rank to measure and estimate the defocus amount. After obtaining sparse defocus map, the above four methods [25–28] employ Laplacian matting guided by the input image to generate the full defocus map. Because textures in the guided image are often transferred into the target image [22, 24], the full defocus map of these methods [21, 23, 25–28] usually suffer from the influence of textures of input image.

To address this problem, some studies are conducted. The authors of [29] use KNN matting [30] for interpolation and introduce a post-processing, where firstly the input image is over-segmented into super pixels and then the defocus amount is averaged within the super pixel. In [31], the input image is also over-segmented into super pixels and defocus amount is estimated for each super pixel using transductive inference. In our previous work [32], structure-texture decomposition is employed to eliminate textures in the guided image. The authors of [33] unify multi-scale deep and hand-crafted features together to estimate defocus amount at edge points and employ Laplacian matting guided by a smooth version of the input image to obtain the full defocus map. These four methods [29, 31–33] can ease the problem to some extent, however, the influence of textures cannot be perfectly eliminated. The authors of [34] propose to use the domain transfer filter (an edge-aware smoothing filter) to avoid textures in the guided image. The estimated defocus map is smooth, however, it may suffer from numerical instability since it uses the division of the filtering results of the sparse defocus map and the edge map as the final defocus map.

2.2. Region-based methods

To avoid the propagation procedure in edge-based methods, the region-based methods try to directly estimate the defocus amount for each image patch. In [35], a circular Radon transform is designed and the defocus amount is estimated in the frequency domain for spatially-invariant defocus blur. In [36], a general regression neural network is proposed to first obtain the blur type and then estimate its parameters. The method is effective, however, it can only handle spatially-invariant defocus blur.

In [37], a sparsity feature, i.e., the sparsity of the local image patch on a defocused patch dictionary, is proposed to measure defocus amount. Their method can only get accurate estimation for small defocus areas since the relationship between the sparsity feature and the defocus amount is not reliable when the defocus amount is large. The authors of [38] use log averaged spectrum residual to obtain a coarse defocus map and refine it iteratively by exploiting the relevance of similar neighbor regions. Since the log averaged spectrum residual is insensitive to defocus amount varying when the defocus amount is large, their defocus map is not reliable for high-level defocus regions. In [39], a novel blur feature based on high frequency of multi-scale local discrete cosine transform is proposed to measure blur amount. The authors of [40] proposes a blur feature based on multi-scale singular value decomposition to measure blur amount. These two methods

[39, 40] can only tell whether a local patch is blurry. Recently, as deep learning can get awesome performance on many computer vision tasks, some researchers try to use it for DME. However, since there is not enough RGB-defocus amount pairs, the deep learning based DME is not truly investigated. Fortunately, benefitting from the CUHK blur detection dataset [41], there are several trials using deep learning for a relaxed problem of DME, i.e., defocus blur detection. The authors of [42] propose an end-to-end defocus detection method based on a multi-stream bottom-top-bottom fully convolutional network. In [43], a fully convolutional neural network with pyramid pooling and boundary refinement layers is employed to generate defocus blur map. In [44], locally relevant features at the super-pixel level are learnt by a multiple convolutional neural networks. Since the training dataset only has two or three levels of blur amount, there are also mainly two [42, 44] or three [43] levels in the blur map.

A framework of localized 2D frequency analysis for spatially-varying blur is first proposed in [18]. A set of orthogonal filters is employed to derive the likelihood for certain blur kernels. Based on this framework, the authors of [18] use rectangular filter windows to estimate motion blur. In [9], Gabor filters, whose filter windows are Gaussian windows, are employed to estimate defocus blur and the authors find that the true defocus radius r is usually caught by the first several local maximums of the likelihood. Based on this discovery, they took the local maximums as coarse defocus maps and developed a coherent labeling procedure to refine them. In [10], the likelihoods for a predefined candidate set of blur kernels were calculated and sorted from high to low. Then the order of the candidates was sent to an RTF as input feature to obtain the final defocus map. They also established a small realistic defocus dataset in the sense of image deblurring with the help of a light field camera [14]. Usually, the defocus map estimated via localized frequency analysis tends to enlarge the slightly defocused area and therefore the discontinuity of the defocus map does not always coincide with the real defocus discontinuity.

2.3. Summary

The performance of edge-based methods or region-based methods is good for different parts of the input image: the former performs well for edge regions and defocus discontinuities, while the latter performs well for regions where there are no defocus discontinuities. They are complementary to each other. Therefore, we propose to combine them together, in order to keep their respective advantages while eliminating the shortcomings. In contrast to deep learning-based methods, there are two advantages for the proposed method: firstly, the proposed method is much more interpretable than deep learning-based methods; and secondly, unlike deep learning-based methods, the proposed method does not need a large number of training samples to train the model.

3. Proposed method

The framework of the proposed method is shown in Figure 1. The defocus map is obtained via an RTF, of which there are two very important elements: the input feature and the linear basis. The input image is processed via localized 2D frequency analysis to obtain the likelihood function $L(r)$ with respect to defocus amount r, and then the first K highest local maximums of the likelihood are taken as the input feature of RTF. At the same time, the estimations of some edge-based DME methods are calculated and serve as the linear basis of RTF. The details are described as follows.

3.1. Localized 2D frequency analysis for likelihood

Frequency spectrum of the input image is a natural cue to estimate the defocus amount. Localized 2D frequency in [18] and [9] can be used to estimate the likelihood that a small image patch is blurred with a candidate blur kernel. However, in the derivation of their analysis, the orientations of the gradient operator and the frequency analysis filters are not fully considered. To understand this, in this subsection we start with the localized 2D frequency analysis in [18] and [9].

The assumptions for localized 2D frequency analysis are listed as follows. (i) The point spread function (PSF) of the camera can be parameterized by defocus amount r as $h(r)[n]$, where n is the location of image pixel. (ii) The captured image $y[n]$ can be viewed as the convolution of the latent all-sharp image $x[n]$ and PSF, corrupted by additive noise $z[n]$, i.e., $y[n] = x[n] \otimes h(r_n)[n] + z[n]$. (iii) In any small local window $w[n]$ of size $N \times N$, the defocus amount r_n keeps the same. (iv) In any local window $w[n]$, the gradient of the latent image $x^{\nabla}[n]$ is independent and identically distributed (i.i.d.) following a Gaussian distribution, i.e., $x^{\nabla}[n] \overset{i.i.d.}{\sim} \mathcal{N}(0, s^{\nabla})$, where $\mathcal{N}(0, s^{\nabla})$ is the isotropic zero-mean Gaussian distribution whose variance is s^{∇}. (v) For the whole image, the additive noise $z[n]$ is Gaussian white, i.e., $z[n] \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_z^2)$.

The localized 2D frequency analysis is conducted by applying a set of Gabor filters $\{f_i[n]\}_i$ to the gradient of the input image $y^{\nabla}[n]$. In [18], it is proven that the filtering results $\{y_i^{\nabla}[n]\}_i$ are jointly Gaussian distributed and that each dimension is uncorrelated with the rest dimensions. Therefore, the joint distribution of the filtering results can be written as the production of the marginal distributions:

$$p\left(\{y_i^{\nabla}[n]\}_i \left| h(r), s^{\nabla}\right.\right) = \prod_i \mathcal{N}\left(y_i^{\nabla}[n] \left| 0, s^{\nabla}\sigma_{hi}^2 + \sigma_{zi}^2\right.\right), \tag{1}$$

where s^{∇} is the variance of the gradient of the all-sharp image (which can be estimated via fixed point iteration as [9, 18] did), $\sigma_{hi}^2 = \sum_n \left|(h(r) \otimes f_i)[n]\right|^2$ and $\sigma_{zi}^2 = \sigma_z^2 \sum_n \left|(\nabla \otimes f_i)[n]\right|^2$ denote the blur spectrum and the noise spectrum (σ_z^2 is the variance of noise and ∇ is the gradient filter), respectively. Then the log likelihood $L_{\nabla}(r)$ can be calculated from the following equation:

$$\begin{aligned} L_{\nabla}(r) &= \ln p\left(\{y_i^{\nabla}\}_i \left| h(r), s^{\nabla}\right.\right) \\ &= \sum_i \left(-\frac{1}{2}\ln\left(2\pi\left(s^{\nabla}\sigma_{hi}^2 + \sigma_{zi}^2\right)\right) - \frac{\left|y_i^{\nabla}\right|^2}{2\left(s^{\nabla}\sigma_{hi}^2 + \sigma_{zi}^2\right)} \right). \end{aligned} \tag{2}$$

Here, $L_{\nabla}(r)$ and y_i^{∇} vary over pixel location n and are abbreviated for convenience and simplification.

Now let us analyze each term in the likelihood. The first term in Equation (2) comes from the normalization factor of the Gaussian distribution and the second term comes from the shape factor. For a Gaussian distribution, usually, the shape factor contains much more information than the normalization factor. Consequently, the second term can often provide more information than the first term. Please note that σ_{hi}^2 and σ_{zi}^2 are independent of the input image $y[n]$. Since different image patches can have the same s^{∇}, the first term can only provide very limited information to the likelihood. In contrast, $y_i^{\nabla}[n]$ is highly dependent on image content and usually differs from patch to patch. Therefore, the information contributed by applying filter $f_i[n]$ to image $y[n]$ is mainly provided by the second term.

In the derivation of [9], the authors use the gradient operator of 0° and a Gabor filter set with all possible orientations. However, the response would be less meaningful and unreliable when the orientations of the gradient operator and the Gabor filter are orthogonal. Please note the fact that for any image patch, if it is filtered by two filters with orthogonal orientations, the response will be very small. Mathematically, $y_i^{\nabla_{0^\circ}}[n] \to 0$ when $ori(f_i) \to 90^\circ$. Consequently, the second term of (2) will go to 0 when $ori(f_i) \to 90^\circ$, which means that the filters whose orientations are close to 90° can hardly provide any useful information in their case. Therefore, in order to avoid small responses, we propose to use the Gabor filters whose orientations are close to the gradient operator ∇_{0°. However, as a result, the filters whose orientations are close to 90° are not used.

To make full use of the Gabor filter set, we employ two gradient operators with orthogonal orientations, i.e., ∇_{0° and ∇_{90°. To avoid small filter responses, we only employ the Gabor filters whose orientations are in $[0^\circ, 45^\circ] \cup [135^\circ, 180^\circ]$ for gradient ∇_{0° and the filters whose orientations are in $[45^\circ, 135^\circ]$ for gradient ∇_{90°. It can be easily verified that for ∇_{0° and ∇_{90°, the frequency spectrums are uncorrelated. Therefore, the joint distribution of $\{y_i^{\nabla_{0^\circ}}[n]\}_i$ and $\{y_i^{\nabla_{90^\circ}}[n]\}_i$ can be written as the production of the two marginal distributions. Consequently, the (log) likelihood can be written as follows:

$$L(r) = L_{\nabla_{0^\circ}}(r) + L_{\nabla_{90^\circ}}(r). \tag{3}$$

Therefore, the likelihood for each orientation can be calculated separately and summed up to obtain the final likelihood. The whole workflow is illustrated in Figure 2. Now the likelihood that an image patch is defocused by the blur kernel $h(r)$ is obtained. Then the defocus amount r can be directly estimated via maximum likelihood (ML).

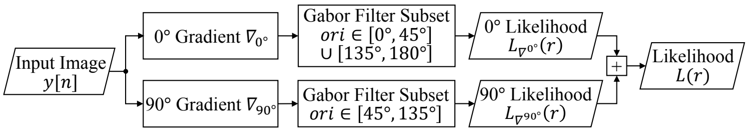

Input Image
$y[n]$
0° Gradient ∇_{0°
90° Gradient ∇_{90°
Gabor Filter Subset
$ori \in [0°, 45°] \cup [135°, 180°]$
Gabor Filter Subset
$ori \in [45°, 135°]$
0° Likelihood
$L_{\nabla^{0^\circ}}(r)$
90° Likelihood
$L_{\nabla^{90^\circ}}(r)$
+
Likelihood
$L(r)$

Figure 2: Workflow of the localized 2D frequency analysis.

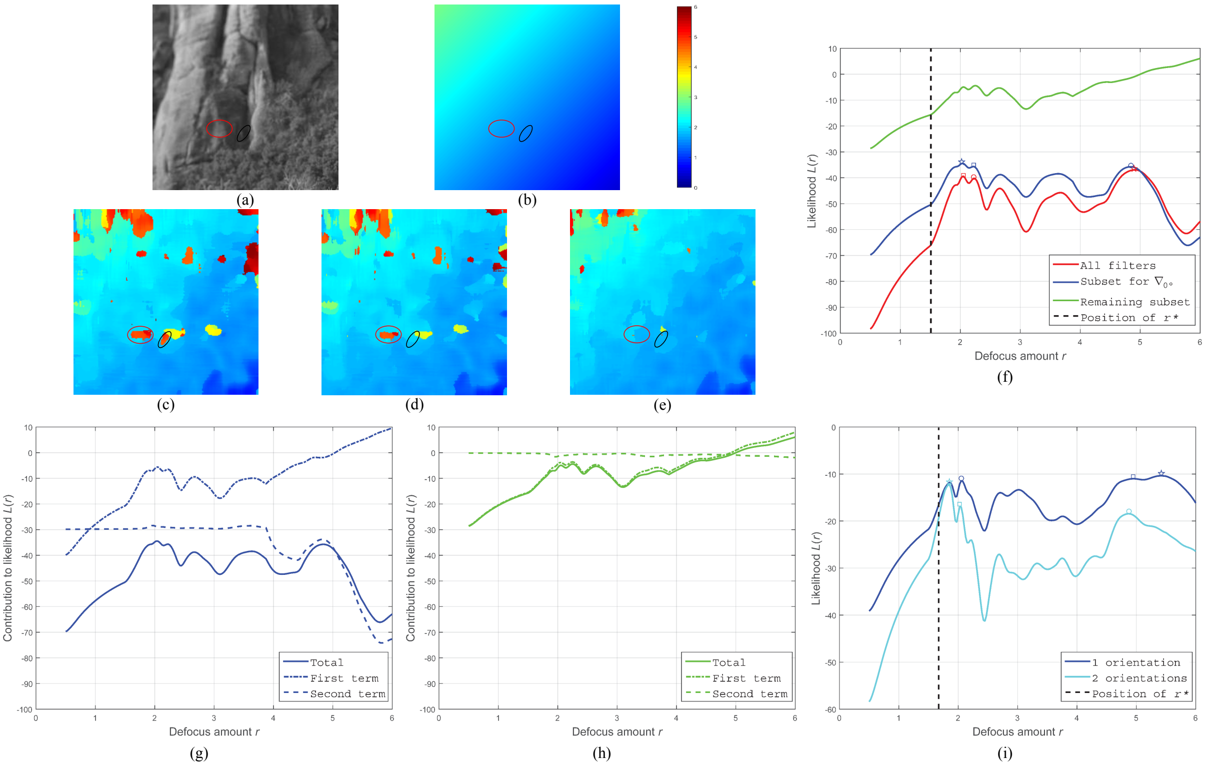


Figure 3: Example of using two orthogonal gradients and the corresponding subsets of Gabor filters: gradually varying defocus blur. (a) Input image. (b) Ground truth defocus map. (c) ML result, using ∇_{0° with all Gabor filters. (d) ML result, using ∇_{0° with the selected subset of Gabor filters. (e) ML result, using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters. (f) Likelihood calculated at pixel (166,120), which is the center of the black ellipse in (a-e). (g) Detailed likelihood for the selected subset at pixel (166,120). (h) Detailed likelihood for the remaining subset at pixel (166,120). (i) Likelihood calculated at pixel (161,94), which is the center of the red ellipse in (a-e). For (f) and (i), the first 3 highest local maximums are marked as pentagram(☆), square(□) and circle(◯), respectively. See the text for details.

An example demonstrating the advantage of using two orthogonal gradients with the corresponding subsets of Gabor filters is shown in Figure 3, where the input image and the ground truth defocus map are given in (a) and (b) respectively. In this example, the image is distorted by gradually varying defocus blur. We directly conduct ML on the likelihoods estimated by (i) using ∇_{0° with all Gabor filters, (ii) using ∇_{0° with the corresponding selected subset of Gabor filters, and (iii) using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters. The estimated defocus maps are shown in (c), (d) and (e) respectively. It can be

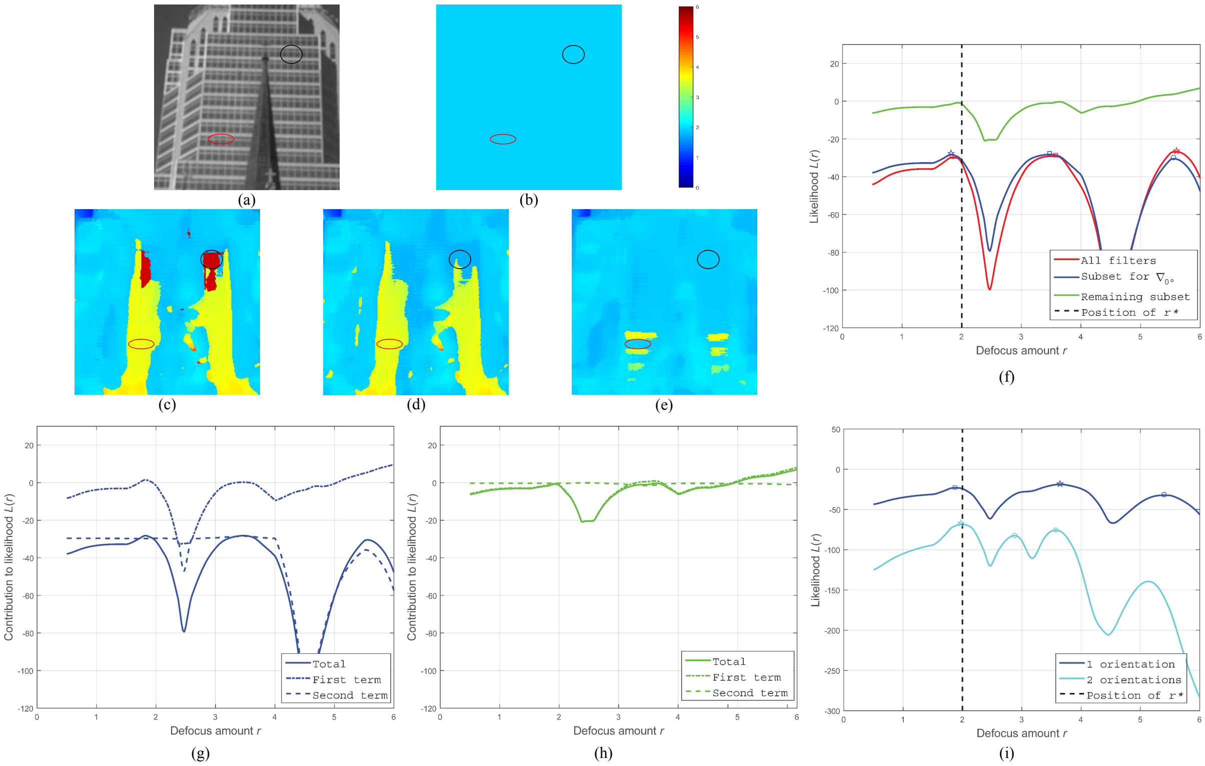


Figure 4: Example of using two orthogonal gradients and the corresponding subsets of Gabor filters: uniform defocus blur. (a) Input image. (b) Ground truth defocus map. (c) ML result, using ∇_{0° with all Gabor filters. (d) ML result, using ∇_{0° with the selected subset of Gabor filters. (e) ML result, using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters. (f) Likelihood calculated at pixel (69,178), which is the center of the black ellipse in (a-e). (g) Detailed likelihood for the selected subset at pixel (69,178). (h) Detailed likelihood for the remaining subset at pixel (69,178). (i) Likelihood calculated at pixel (171,87), which is the center of the red ellipse in (a-e). For (f) and (i), the first 3 highest local maximums are marked as pentagram(☆), square(□) and circle(◯), respectively.

easily found that using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters can obtain the most accurate estimation. Figure 3 (f) gives the detailed likelihood of the pixel at the center of the black ellipse in (a-e), where using the corresponding selected subset of Gabor filters can get better estimation than using all Gabor filters. The first three local maximums of the likelihood are marked as pentagram(☆), square(□) and circle(◯), respectively. As can be seen, the first local maximum when using the selected subset is close to the true defocus amount r^*, while when using all filters the first local maximum is extremely larger than r^*. To appreciate this, we plot the detailed likelihoods for the selected and remaining subsets, shown in (g) and (h) respectively. It can be found that the dash-dot lines in (g) and (h) are quite similar. That is to say, the likelihoods contributed by the first term of Equation (2) are very similar for the selected and remaining subsets (the difference of the range is caused by the different number of filters in the subsets). In other words, the first term can hardly provide any useful information. On the other hand, the dash line in (g) is far from 0 and fluctuates notably with r, meaning that the second term can provide useful information for the selected subset. However, the dash line in (h) is very close to 0, meaning that the second term cannot provide any useful information for the remaining subset. Therefore, for using all filters, the useful information provided by the selected subset can be concealed by the remaining subset. Now we can conclude that using the selected subset filters can get more accurate likelihood because the remaining subset filters can provide no useful information but undesirable noise. Figure 3 (i) shows the detailed likelihood of the pixel at the center of the red ellipse in (a-e), where using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters can get better estimation than using only ∇_{0°. As can be seen, using two orientations can catch the true defocus amount r^* by the first local maximum (marked as pentagram(☆)), while using one orientation cannot catch r^* by any of the first three highest local maximums. The reason is: $L_{\nabla_{0^\circ}}(r)$ and $L_{\nabla_{90^\circ}}(r)$ can provide complementary information, therefore, using ∇_{0° and ∇_{90° together can obtain more accurate likelihood.

We also conduct demonstration experiments on images distorted by uniform defocus blur and piecewise uniform defocus blur. The results are shown in Figures 4 and 5. These experiments can also provide similar conclusions to that for gradually varying defocus blur. These three experiments demonstrate that the proposed two-orientation localized 2D frequency analysis is effective and can obtain more accurate likelihood than the original one-orientation localized 2D frequency analysis. As shown by Figures 3-5, the ML estimations are insensitive to textures of the input image. This owes to the advantage of the region-based approach, i.e., using 2D localized frequency analysis to extract defocus information from defocused regions.

3.2. Likelihood feature

As demonstrated in the last subsection, the likelihood can be used to estimate the defocus map via ML directly. However, due to noise and calculation errors, the second highest local maximum can surmount the latent maximum, which is the maximum for ideal condition. Consequently, the ML estimation can be unreliable. Therefore, to obtain more accurate defocus map, we propose to extract a proper feature from the likelihood and refine it with an RTF. It should be noted that the extracted feature will be insensitive to textures of the input image since the likelihood is calculated in a region-based way, i.e., via 2D localized frequency analysis.

Intuitively, the likelihood itself can be directly selected as the input feature. However, it contains too much redundant information to train the RTF. Further, the dimension is so high that the training of the RTF will be too time-consuming. Moreover, the ranges of the likelihood are different for different images and need to be normalized properly. Therefore, extracting a more expressive feature from the likelihood is necessary.

A feature extracted by uniformly discrete sampling of likelihood is employed in [10]. Specifically, the order of the sampling points, which are sorted by their likelihoods from high to low, serves as the input feature.

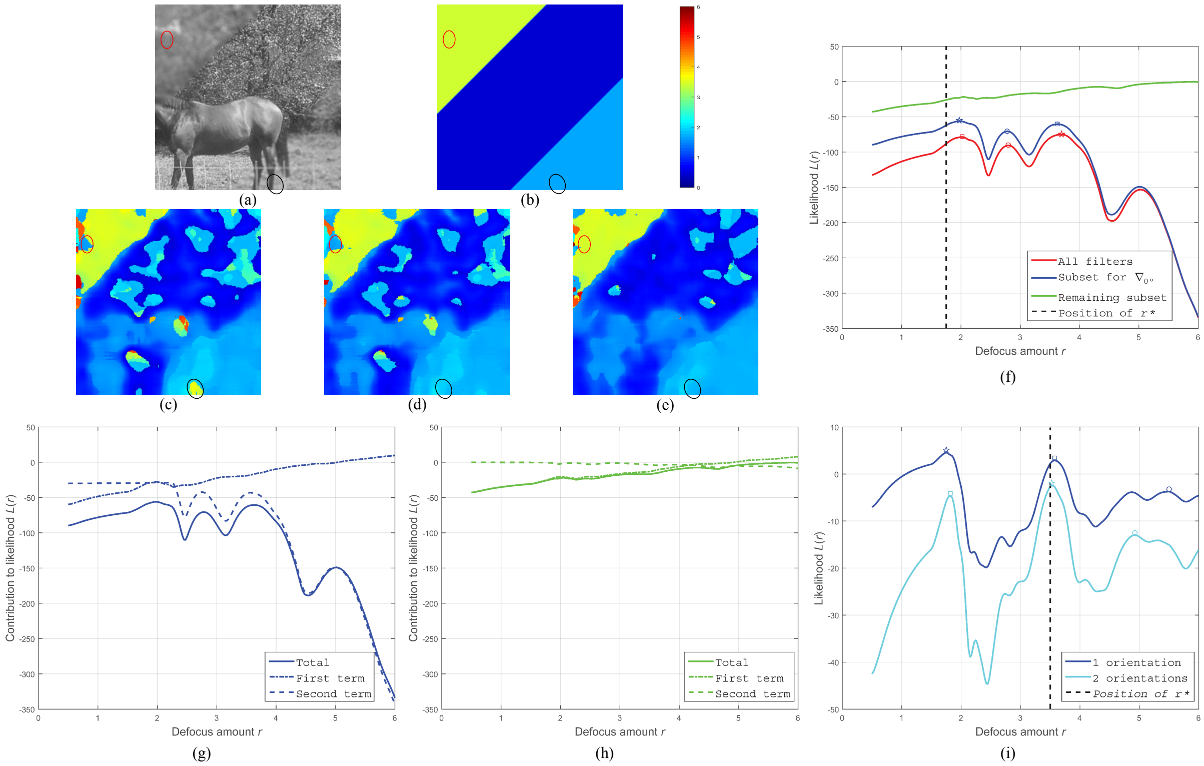


Figure 5: Example of using two orthogonal gradients and the corresponding subsets of Gabor filters: piecewise uniform defocus blur. (a) Input image. (b) Ground truth defocus map. (c) ML result, using ∇_{0° with all Gabor filters. (d) ML result, using ∇_{0° with the selected subset of Gabor filters. (e) ML result, using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters. (f) Likelihood calculated at pixel (235,153), which is the center of the black ellipse in (a-e). (g) Detailed likelihood for the selected subset at pixel (235,153). (h) Detailed likelihood for the remaining subset at pixel (235,153). (i) Likelihood calculated at pixel (46,18), which is the center of the red ellipse in (a-e). For (f) and (i), the first 3 highest local maximums are marked as pentagram(☆), square(□) and circle(◯), respectively.

However, since sampling points that are far from local maximums cannot provide any useful information, this feature is also redundant and needs to be further refined. Therefore, to reduce the redundant information, we take the first K highest local maximums instead of the order of sorted sampling points as the input feature. The motivation is that the first several highest local maximums of likelihood can catch the true defocus amount with a high probability. This can reduce the feature dimension while catching all the essential information. For each pixel, there will be K local maximums and thus the feature is a K-channel feature. When there is not enough local maximums, the shortage is complemented via copying the lowest local maximum.

We briefly justify the proposed feature by comparing it with the feature of [10] on three synthetically defocused images, which are blurred by uniform defocus, gradually varying defocus and piecewise uniform defocus with sharp discontinuities. Since the features are to be refined via RTF, we evaluate them in the

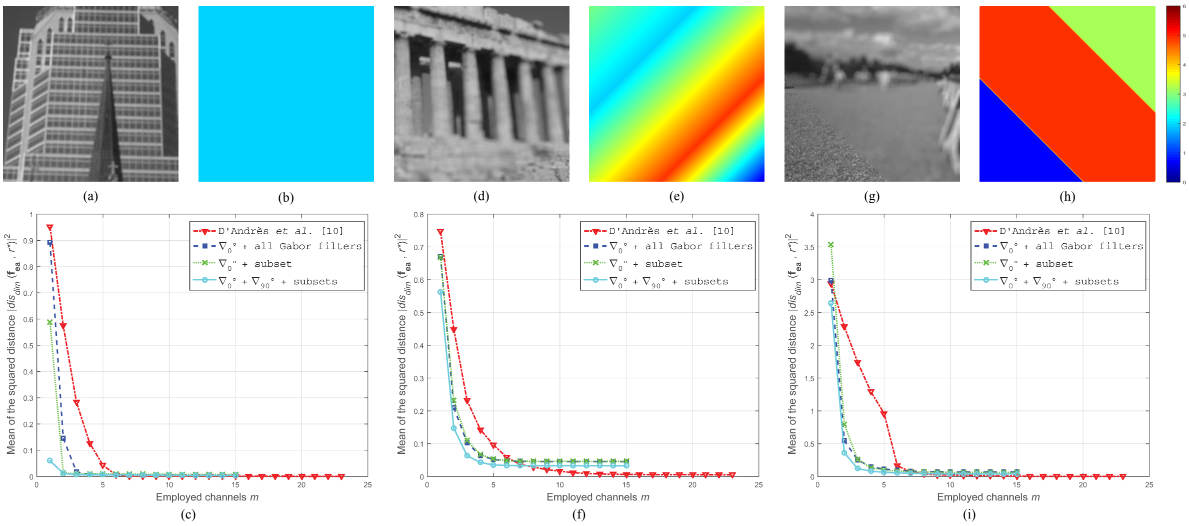


Figure 6: Effectiveness of the likelihood feature. (a) Input image blurred by uniform defocus. (b) Ground truth. (c) Mean of the squared distance (MSD). (d) Input image blurred by gradually varying defocus. (e) Ground truth. (f) MSD. (g) Input image blurred by piecewise uniform defocus with sharp discontinuities. (h) Ground truth. (i) MSD.

perspective of refinement. Therefore, the metric is selected as the mean of the squared distance (MSD) from the feature $\mathbf{f_{ea}}[n]$ to the ground truth defocus amount $r^*[n]$. Here, the distance is defined as

$$dis_m(\mathbf{f_{ea}}[n], r^*[n]) = \min_{i \leq m} |f_{ea}(i)[n] - r^*[n]|, \tag{4}$$

where $f_{ea}(i)[n]$ is the i-th channel of the extracted feature at pixel n, m is the number of the employed channels and $|\cdot|$ denotes the absolute value. After the distance is calculated at each pixel, the MSD can be obtained as the average of the squared distances. The comparison results are shown in Figure 6, where the input images are given in (a), (d) and (g), and the ground truth are provided in (b), (e) and (h). To show the effect of different choices of likelihood, we provide three versions of likelihood calculated by using ∇_{0° with all Gabor filters, ∇_{0° with the selected subset of Gabor filters, and ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters, respectively. They are shown as blue long dash line, green short dash line, and cyan solid line in Figure 6.

From (c), (f) and (i), it can be found that for all the four features the MSD decreases very slowly when m is large. Therefore, the most proper dimension of the feature can be decided according to the MSD curve. As can be seen, for the three local maximum based features, the one calculated by using two-orientation localized 2D frequency analysis can achieve the lowest MSD. This demonstrates again that the two-orientation localized 2D frequency analysis is effective and can obtain more accurate likelihood. The MSD of the proposed feature decreases faster and is small enough in all the three cases when $m = 5$, while for the feature of [10] m needs to be 7 to achieve the same MSD. That is to say, the proposed feature can catch the essential information with a lower dimension. It should be pointed out that though the feature

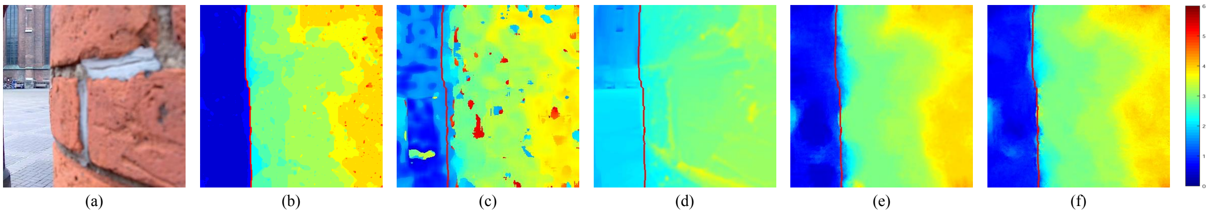


Figure 7: Effect of using edge-based methods as linear basis. (a) Input image. (b) Ground truth defocus map. (c) The first channel of the proposed feature. (d) Defocus map estimated by [32]. (e) Refined defocus map via RTF without linear basis. (f) Refined defocus map via RTF with linear basis.

of [10] can achieve lower MSD, it sacrifices the feature dimensionality for MSD. The possible lowest MSD depends on the length of sampling interval: the shorter the interval, the lower the MSD. In the extreme case where the sampling interval approximates to 0, the MSD will also approximate to 0. However, the feature dimension will be infinity at the same time. Among the numerous channels of the feature, only a few are effective, since the rest channels are far from the true defocus amount. Therefore, the feature of [10] contains much redundant information and needs a higher dimension to catch essential information than the proposed feature (the first K highest local maximums of likelihood calculated by two-orientation localized 2D frequency analysis).

Once extracted, the proposed feature can be directly sent to an RTF to obtain a defocus map.

3.3. Combination with edge-based methods via RTF

As mentioned in Section 2, region-based methods cannot catch defocus discontinuities very well, while edge-based methods can. The proposed feature in the last subsection is region-based and may possibly not catch defocus discontinuities very well, see Figure 7 for example, where the input images and ground truth are shown in (a) and (b) respectively. The first channel of the proposed feature is provided in Figure 7 (c): As can be seen, it cannot catch the blur discontinuity denoted by the red line. If sending the feature directly to an RTF, we can obtain a refined defocus map, as shown in Figure 7 (e). The refined map is much better than the input feature, however, it still cannot catch the defocus discontinuity very well. Fortunately, edge-based methods can catch the defocus discontinuity well, as shown in Figure 7 (d), where the result of our previous work [32] is provided as an example. Therefore, to solve the aforementioned problem, we propose to incorporate edge-based methods into our method by taking them as the linear basis of the RTF.

Introduced by [17], RTF is a specially-designed Gaussian conditional random field to predict the label image $\mathbf{d}$ (noted purposely to match the notation of a defocus map) of an input image $\mathbf{y}$ in a globally consistent way. The energy function is defined as the sum of local potentials:

$$E\left(\mathbf{d}|\mathbf{y}\right)=\sum_{t}\sum_{F\in\mathcal{F}_t}\frac{1}{2}\mathbf{d}_F^T\mathbf{Q}_t\left(\mathbf{y}_F\right)\mathbf{d}_F-\mathbf{d}_F^T\mathbf{W}_t\left(\mathbf{y}_F\right)\mathbf{b}_t\left(\mathbf{y}_F\right). \tag{5}$$

Here t is the factor type and $\mathcal{F}_t$ is the set of all factors with type t. $\mathbf{d}_F$ and $\mathbf{y}_F$ denote the label image and input image at pixels related to factor F, respectively. $\mathbf{Q}_t\left(\mathbf{y}_F\right)$ and $\mathbf{W}_t\left(\mathbf{y}_F\right)$ are the parameters of RTF, which are determined by the input image $\mathbf{y}$ in a regression tree manner. $\mathbf{b}_t\left(\mathbf{y}_F\right)$ is the linear basis. The only hyper parameter to be tuned is the maximum depth of the regression trees, which makes the RTF non-parametric. Once the global energy function is calculated, the predicted defocus map $\mathbf{d}^*$ can be written as

$$\mathbf{d}^* = \left[\mathbf{Q}(\mathbf{y})\right]^{-1} \mathbf{l}(\mathbf{y}), \tag{6}$$

where $\mathbf{Q}(\mathbf{y}) = \sum_t \sum_{F\in\mathcal{F}_t} \mathbf{Q}_t\left(\mathbf{y}_F\right)$ and $\mathbf{l}(\mathbf{y}) = \sum_t \sum_{F\in\mathcal{F}_t} \mathbf{W}_t\left(\mathbf{y}_F\right) \mathbf{b}_t\left(\mathbf{y}_F\right)$ are the quadratic matrix and the linear term, respectively. Since $\mathbf{Q}(\mathbf{y})$ is a symmetric positive definite matrix, the inference can be solved via conjugate gradient very fast once $\mathbf{Q}(\mathbf{y})$ and $\mathbf{l}(\mathbf{y})$ are obtained.

Let us now focus on the linear basis $\mathbf{b}_t(\mathbf{y}_F)$ of RTF. As demonstrated by the application of image denoising in [45], using linear basis can integrate the complementary information from the basis into the input feature, to obtain better results. The simplest choice for linear basis is a **1** vector whose elements are 1, which is employed by [10] in their DME method. However, for this choice, no complementary information is incorporated. In this work, to make full use of the complementary information from edge-based methods, we employ the results of three edge-based methods as the linear basis, instead of a **1** vector. The methods of [20, 21, 32] are selected as the linear basis since they are very typical edge-based methods and can catch defocus discontinuities reasonably well. The results of these three methods are directly concatenated together as a three-channel linear basis.

Through this improvement, the aforementioned problem can be solved and the refined defocus map via RTF with linear basis is shown in Figure 7 (f). From (e) and (f), we can find that using edge-based methods as linear basis can obtain sharper discontinuities at correct locations. This means that the complementary information has been successfully integrated into the input feature.

4. Experiments

Our experiments are conducted on the synthetic and realistic datasets generated by [10]. The synthetic dataset contains 125 synthetically defocused gray images of size 239 × 239. The realistic dataset contains 22 RGB images of size 360 × 360, captured with a light field camera. The proposed method is compared with several state-of-the-art DME methods, including [10, 21, 32–34, 37]. Among these methods, [21, 32–34] are edge-based methods, and [10, 37] are region-based methods. All the methods are conducted with their default settings. It should be pointed out that the PSFs employed in [21, 32–34, 37] are in the form of isotropic 2D Gaussian function and the PSF used in [10] is in the form of disk function. To compare these methods with the proposed method, a conversion (suggested by [34]) from the standard deviation of Gaussian PSF to the radius of disk PSF is conducted.

4.1. Implementation details

In the implementation of the proposed method, K is set to 5 to catch all essential information while avoiding redundant information. Following the settings of [10], all other parameters are set as follows. The defocus kernel is in the form of disk function and the range of defocus amount is $[0.5, 6]$[1]. In the localized 2D frequency analysis, the window size N is set to 41 and the Gabor filters are the same with [9]. The neighborhood connectivity of the RTF is 5×5 (i.e., 1 unary tree with 12 pairwise trees, whose offsets are [-2,-2], [-1,-2], [0,-2], [-2,-1], [-1,-1], [0,-1], [-2,0], [-1,0], [-2,1], [-1,1], [-2,2] and [-1,2]). The stacked depth of RTF D_S is 3 and the maximum depth of each tree D_T is 10. The RTF is trained to minimize the mean square error (MSE) on the same 100 images with [10], which are from the synthetic dataset, and tested on the rest 25 synthetic images as well as the realistic dataset.

The synthetic dataset comes from [10]. It is generated by synthetically defocusing some all-in-focus images with predefined defocus maps. The all-in-focus images are cropped from the all-in-focus areas of the images in the Berkeley segmentation dataset [46]. The predefined defocus maps are of uniform, piecewise uniform and gradually varying formulations, respectively. The range of defocus amount in the defocus maps is $[0.5, 6]$ and the PSF is in the form of disk function.

4.2. Ablation studies

To show the effect of each hyper parameter, ablation studies are conducted on the synthetic dataset.

Local window size N. N is analyzed for $\{21, 31, 41, 51, 61\}$ on the whole synthetic dataset. The likelihoods are calculated and the defocus map is estimated via ML. The prediction errors are shown in Table 1, where the quantitative metrics are mean absolute error (MAE) and MSE, for which the lower the better. As can be seen, the errors first decrease and then increase as N increases. The reason is explained as follows. For small local window, the Gabor filter set is also small, therefore, the likelihood might be influenced by noise and calculation errors heavily. For large local window, the assumption that the defocus amount keeps the same in any local window does not hold very well, therefore, the likelihood might be inaccurate. Consequently, N should be chosen appropriately. For example, 41 or 51 is a suitable value.

When conducting ablation studies on the neighborhood connectivity of RTF, the stack depth of RTF D_S and the maximum depth of each tree D_T, we vary one of the three parameters while keeping the other

[1] We have investigated the maximum defocus amount used in the literature and find that they are usually no more than 8 [9]. For middle and low resolution images, the defocus amount 6 is a large defocus blur, and $[0.5, 6]$ can cover almost all situations. For ultra-high resolution images, there might be extremely defocused areas whose defocus amount is larger than 6. If we directly apply existing methods to these areas, perhaps wrong estimations will be produced. Fortunately, in these areas, the problem can be eased to some extent by down-scaling the input image and up-scaling the estimated defocus map afterwards.

Table 1: Ablation study on local window size N.

N	MAE	MSE
	Mean/Std. dev	Mean/Std. dev
21	0.862/0.192	1.459/0.547
31	0.630/0.228	0.897/0.562
41	0.473/**0.188**	0.619/**0.430**
51	**0.457**/0.208	**0.587**/0.462
61	0.470/0.217	0.606/0.495

Table 2: Ablation study on the neighborhood connectivity of RTF.

neighborhood	MAE	MSE
connectivity	Mean/Std. dev	Mean/Std. dev
1 × 1	0.290/0.098	0.292/0.251
3 × 3	0.263/0.097	0.260/0.216
5 × 5	**0.247**/**0.091**	**0.237**/0.213
7 × 7	0.256/0.095	0.252/**0.210**

Table 3: Ablation study on the stack depth of RTF D_S.

D_S	MAE	MSE
	Mean/Std. dev	Mean/Std. dev
1	0.329/0.101	0.321/0.231
2	0.272/0.091	0.260/0.214
3	0.252/0.089	0.245/0.216
4	0.250/**0.088**	0.241/0.206
5	0.242/**0.088**	**0.238**/0.206
6	**0.241**/**0.088**	**0.238**/0.205
7	0.246/**0.088**	0.239/**0.201**

two fixed to the values specified in the last subsection. The RTF is trained on 100 images of the synthetic dataset and tested on the rest 25 images.

Neighborhood connectivity of RTF. The neighborhood connectivity is analyzed for $\{1\times1, 3\times3, 5\times5, 7\times7\}$. The test errors are shown in Table 2. Due to the spatial continuity of images, the defocus map also possesses some spatial continuity, i.e., a pixel usually have similar defocus amount with its neighbor pixels. This character is effective only in a limited size of neighborhood. As can be seen in Table 2, the performance first ameliorates and then distorts as the neighborhood size increases. Therefore, the neighborhood connectivity of RTF should also be selected appropriately and 5×5 is an appropriate choice.

Stack depth of RTF D_S. D_S is analyzed for $\{1, 2, 3, 4, 5, 6, 7\}$. The test errors are shown in Table 3. Theoretically, for a stacked RTF, the deeper stack can be viewed as a refinement of the shallower stack [17]. Therefore, the deeper the RTF is stacked, the more powerful the RTF is. Consequently, better performance would be produced for larger D_S. However, deeper stack means more running time. Moreover, due to over fitting, the performance improvement would decrease as D_S increases, as can be seen in Table 3. Therefore, D_S should not be too small or too large. In this paper, D_S is set to 3 as a tradeoff between performance and running time.

Maximum depth of each tree D_T. D_T is analyzed for $\{8, 9, 10, 11, 12\}$. The test errors are shown in Table 4. As can be seen, the errors first decrease and then increase as D_T increases. In theory, larger maximum depth of a tree means more degree of freedom and thus larger model capacity. However, with

Table 4: Ablation study on the maximum depth of each tree D_T.

D_T	MAE	MSE
	Mean/Std. dev	Mean/Std. dev
8	0.269/0.108	0.258/0.249
9	0.256/0.098	0.246/0.234
10	**0.247/0.091**	**0.237/0.213**
11	0.279/0.107	0.289/0.229
12	0.266/0.097	0.266/**0.213**

Table 5: Test errors on synthetic, clean realistic and noisy realistic datasets.

Method	Synthetic dataset		Clean realistic dataset		Noisy realistic dataset	
	MAE	MSE	MAE	MSE	MAE	MSE
	Mean/Std. dev	Mean/Std. dev	Mean/Std. dev	Mean/Std. dev	Mean/Std. dev	Mean/Std. dev
$\text{ML}_{1\text{ori}}$	0.562/0.166	0.948/0.487	0.400/0.109	0.409/0.169	0.417/0.097	0.442/0.155
D'Andrès *et al.* [10]	0.269/0.098	0.295/0.236	0.196/0.072	0.094/0.086	0.206/0.067	0.099/0.078
$\text{ML}_{2\text{oris}}$	0.555/0.173	0.895/0.524	0.371/0.100	0.298/0.126	0.389/0.086	0.330/0.108
$\text{L}_{2\text{oris}} + \text{RTF}_K$	0.244/0.088	0.278/0.240	0.178/0.067	0.076/0.071	0.199/0.063	0.091/0.072
$\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Matting}$	**0.240/0.087**	0.255/0.227	**0.176**/0.067	**0.073/0.068**	0.197/0.064	0.090/0.072
$\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Basis}$	0.247/0.091	**0.237/0.213**	0.179/**0.066**	**0.073/0.068**	**0.195/0.061**	**0.086/0.067**

limited training data, the model capacity should be carefully chosen to avoid over fitting. From Table 4, we find that 10 is a suitable value for D_T.

Improvements. To show the effect of each improvement, we compare the proposed method with $\text{ML}_{1\text{ori}}$ (ML using ∇_{0° with all Gabor filters), $\text{ML}_{2\text{oris}}$ (ML using ∇_{0° and ∇_{90° with the corresponding subsets of Gabor filters) and $\text{L}_{2\text{oris}}+\text{RTF}_K$ (refine the first K highest local maximums of the two-orientation likelihood via RTF without edge-based basis). Here, the proposed method is denoted as $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Basis}$ (refine the first K highest local maximums of the two-orientation likelihood via RTF with edge-based basis). We also give the results of [10] as the method is also trained and tested on the same dataset. As suggested by the reviewers, we also try using traditional matting instead of edge-based basis (denoted as $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Matting}$). The test is conducted not only on the 25 test images of synthetic dataset, but also on the clean and noisy realistic datasets. The noisy realistic dataset is generated by corrupting the clean version with additive Gaussian white noise whose standard variance $\sigma_z = 1$ for luminance range of $[0, 255]$. The test errors on are shown in Table 5.

As can be seen, $\text{ML}_{2\text{oris}}$ has smaller MAE and MSE than $\text{ML}_{1\text{ori}}$, $\text{L}_{2\text{oris}} + \text{RTF}_K$ has smaller MAE and MSE than $\text{ML}_{2\text{oris}}$, and $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Basis}$ can obtain much smaller MSE than $\text{L}_{2\text{oris}} + \text{RTF}_K$ (though their MAEs are comparable). It means that all the three improvements are effective in reducing the prediction errors. $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Basis}$ also has smaller MAE and MSE than [10], showing the effectiveness of the improvements from another perspective. $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Matting}$ has comparable MAE and MSE with $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Basis}$, and they both have smaller MAE and MSE than $\text{L}_{2\text{oris}} + \text{RTF}_K$, meaning that both matting and using linear basis can improve the performance. It should be noted that for the noisy realistic dataset, $\text{L}_{2\text{oris}} + \text{RTF}_K + \text{Basis}$ has the smallest MAE and MSE, showing its robustness to noise.

Table 6: DME comparison on clean realistic dataset.

Input Image	Zhuo and Sim [21] MAE/MSE	Shi *et al.* [37] MAE/MSE	D'Andrès *et al.* [10] MAE/MSE	Liu *et al.* [32] MAE/MSE	Park *et al.* [33] MAE/MSE	Karaali and Jung [34] MAE/MSE	Proposed Method MAE/MSE
Image 01	0.459/0.286	0.615/0.453	**0.132**/**0.037**	0.329/0.149	0.155/0.043	0.212/.074	0.139/0.040
Image 02	0.734/0.687	0.680/0.639	**0.178**/0.059	0.527/0.361	0.228/0.097	0.328/0.148	0.179/**0.058**
Image 03	0.448/0.261	0.730/0.680	0.216/0.114	0.430/0.248	0.163/**0.043**	0.285/0.114	**0.162**/0.048
Image 04	0.441/0.276	0.570/0.420	**0.162**/**0.046**	0.314/0.171	0.180/0.050	0.236/0.088	0.177/0.065
Image 05	0.429/0.232	0.853/0.909	0.139/0.036	0.502/0.339	**0.118**/**0.025**	0.328/0.152	0.166/0.051
Image 06	0.524/0.345	0.468/0.298	0.153/0.047	0.352/0.205	0.211/0.077	0.247/0.087	**0.146**/**0.045**
Image 07	0.451/0.256	0.672/0.540	0.187/0.090	0.360/0.205	0.220/0.084	0.367/0.189	**0.166**/**0.052**
Image 08	0.961/1.377	0.940/1.042	0.199/0.123	0.849/1.110	0.372/0.219	0.835/1.195	**0.156**/**0.090**
Image 09	0.752/0.603	0.675/0.677	0.214/0.095	0.491/0.283	**0.136**/**0.039**	0.409/0.216	0.137/0.045
Image 10	0.563/0.389	0.451/0.244	0.204/0.066	0.299/0.143	**0.113**/**0.027**	0.181/0.060	0.184/0.056
Image 11	0.802/0.788	0.520/0.376	0.163/0.067	0.544/0.383	0.225/0.075	0.341/0.206	**0.142**/**0.050**
Image 12	0.862/0.884	0.737/0.713	0.158/0.055	0.638/0.546	0.275/0.114	0.384/0.216	**0.137**/**0.035**
Image 13	0.479/0.306	0.597/0.453	**0.144**/**0.036**	0.427/0.253	0.175/0.060	0.283/0.108	0.182/0.052
Image 14	0.685/0.614	0.779/1.040	0.472/0.398	0.608/0.491	0.490/0.378	0.515/0.410	**0.453**/**0.357**
Image 15	0.935/1.009	0.445/0.263	0.191/0.079	0.647/0.518	0.246/0.089	0.623/0.476	**0.130**/**0.041**
Image 16	0.526/0.408	0.535/0.373	0.196/0.072	0.380/0.268	**0.177**/**0.052**	0.320/0.145	0.187/0.062
Image 17	0.897/0.921	0.566/0.457	0.171/0.057	0.637/0.513	0.407/0.349	0.508/0.347	**0.169**/**0.055**
Image 18	0.636/0.508	0.663/0.517	0.156/0.050	0.479/0.304	0.173/0.049	0.322/0.159	**0.135**/**0.036**
Image 19	0.681/0.570	0.915/1.116	0.317/0.291	0.695/0.623	**0.194**/**0.074**	0.509/0.356	0.239/0.144
Image 20	0.665/0.562	0.507/0.333	**0.185**/0.074	0.540/0.364	0.194/**0.060**	0.287/0.126	0.189/0.094
Image 21	0.672/0.527	0.671/0.662	0.181/0.074	0.450/0.287	**0.167**/**0.044**	0.237/0.090	0.171/0.058
Image 22	0.956/1.019	0.876/0.990	0.200/0.107	0.880/0.910	0.328/0.188	0.409/0.264	**0.196**/**0.084**
Mean	0.662/0.583	0.658/0.600	0.196/0.094	0.517/0.394	0.225/0.102	0.371/0.238	**0.179**/**0.073**
Std. dev	0.182/0.308	0.148/0.271	0.072/0.086	0.161/0.242	0.096/0.097	0.152/0.242	**0.066**/**0.068**

4.3. Comparison on realistic dataset

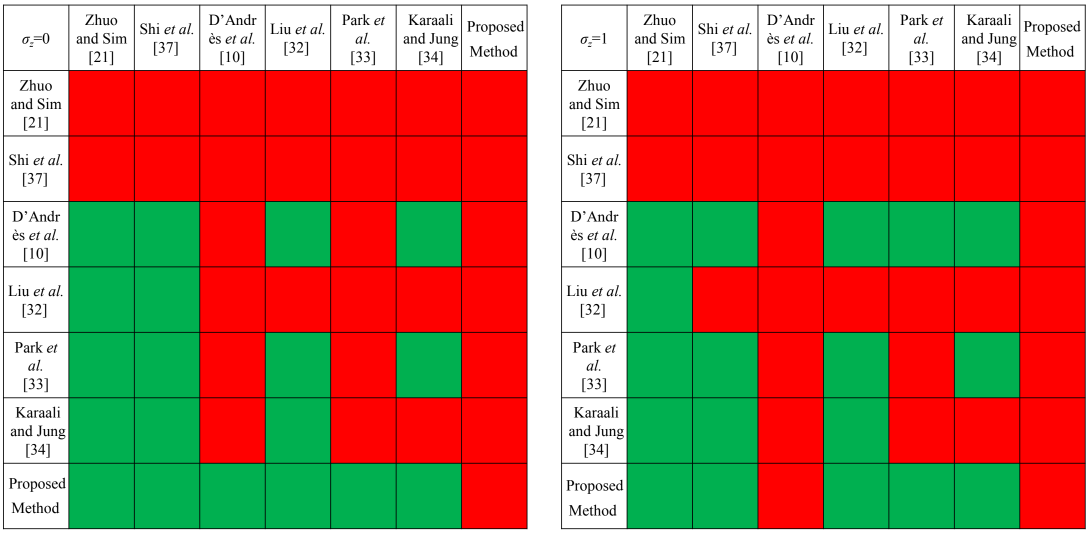
The prediction errors on the realistic dataset are shown in Table 6 (clean version) and Table 7 (noisy version, $\sigma_z = 1$), where the best prediction is in bold. It can be easily found that the proposed method can produce the smallest MAE and MSE on both clear and noisy input images.

To further analyze the quantitative results, we conduct statistical significance tests on the data of Tables 6 and 7. Since the distribution of the data might not be Gaussian distribution, a non-parametric test is adopted: we employ the Friedman test and the results are shown in Figure 8, where (a) is for the clean realistic dataset and (b) is for the noisy version. For each cell, if the row index method is significantly better than the column index method, it is marked as green; otherwise, it is marked as red. It can be found that the proposed method is significantly better than state-of-the art methods on clean dataset. For noisy dataset, the best two methods — [10] and the proposed method — are comparable.

Figure 9 gives visual results on four images from the realistic dataset (two without noise and two with noise). The input images and ground truths are provided in Figure 9 (a) and (b) respectively. It can be found from (c) and (f) that both the methods of [21] and [32] can catch the defocus discontinuities (see the second and third images in (c) and (f)) but suffer from textures of the input image (see the areas of the wall for the first image, the tree trunk for the second and third images and the stones for the forth image). The method of [37] can get correct estimations for slightly defocused areas but often fails with a constant estimation for middle and high levels of defocus areas (see all the four images in Figure 9 (d)). The method of [33] uses a smooth version of the input image as the guidance of Laplacian matting, therefore, it

Table 7: DME comparison on noisy realistic dataset ($\sigma_z = 1$).

Input Image	Zhuo and Sim [21] MAE/MSE	Shi *et al.* [37] MAE/MSE	D'Andrès *et al.* [10] MAE/MSE	Liu *et al.* [32] MAE/MSE	Park *et al.* [33] MAE/MSE	Karaali and Jung [34] MAE/MSE	Proposed Method MAE/MSE
Image 01	0.475/0.309	0.577/0.413	0.183/0.080	0.370/0.197	0.218/0.110	0.206/0.072	**0.161**/**0.056**
Image 02	0.743/0.718	0.565/0.431	0.189/0.066	0.547/0.387	0.232/0.109	0.305/0.134	**0.180**/**0.056**
Image 03	0.460/0.272	0.528/0.366	0.228/0.143	0.434/0.242	**0.155**/**0.038**	0.290/0.118	0.211/0.118
Image 04	0.438/0.266	0.553/0.420	0.164/**0.049**	0.315/0.175	0.210/0.082	0.228/0.083	**0.163**/0.053
Image 05	0.421/0.224	0.638/0.479	0.155/0.042	0.477/0.303	**0.126**/**0.029**	0.337/0.161	0.161/0.044
Image 06	0.534/0.358	0.445/0.270	**0.177**/**0.067**	0.381/0.225	0.236/0.104	0.247/0.087	0.181/0.073
Image 07	0.451/0.253	0.648/0.511	0.213/0.130	0.373/0.206	0.233/0.093	0.364/0.186	**0.191**/**0.066**
Image 08	1.014/1.549	0.825/0.919	**0.206**/**0.111**	0.992/1.565	0.454/0.302	0.871/1.288	0.217/0.136
Image 09	0.758/0.619	0.649/0.661	0.213/0.096	0.521/0.315	**0.143**/**0.042**	0.416/0.224	**0.143**/0.046
Image 10	0.570/0.402	0.387/0.185	0.207/0.070	0.350/0.195	**0.136**/**0.037**	0.189/0.064	0.213/0.085
Image 11	0.811/0.807	0.483/0.320	0.146/0.056	0.571/0.419	0.233/0.081	0.331/0.190	**0.139**/**0.050**
Image 12	0.887/0.943	0.710/0.667	0.160/0.053	0.736/0.742	0.311/0.147	0.393/0.230	**0.153**/**0.047**
Image 13	0.497/0.337	0.572/0.411	**0.153**/**0.041**	0.473/0.306	0.212/0.088	0.290/0.113	0.176/0.050
Image 14	0.694/0.629	0.760/0.997	0.475/0.417	0.620/0.502	0.496/0.390	0.515/0.412	**0.444**/**0.363**
Image 15	0.955/1.059	0.478/0.318	0.215/**0.099**	0.705/0.630	0.349/0.177	0.641/0.512	**0.198**/0.106
Image 16	0.553/0.471	0.473/0.315	0.199/**0.067**	0.439/0.391	0.230/0.087	0.325/0.149	**0.198**/0.071
Image 17	0.921/0.974	0.529/0.392	**0.181**/**0.062**	0.756/0.712	0.507/0.453	0.541/0.386	0.196/0.070
Image 18	0.651/0.536	0.684/0.566	0.191/0.096	0.523/0.372	0.248/0.134	0.320/0.158	**0.154**/**0.069**
Image 19	0.694/0.593	0.917/1.120	0.277/0.153	0.710/0.637	0.247/0.167	0.514/0.360	**0.225**/**0.091**
Image 20	0.693/0.619	0.494/0.332	**0.186**/**0.075**	0.609/0.491	0.225/0.088	0.300/0.136	0.203/0.108
Image 21	0.676/0.533	0.651/0.625	0.186/0.082	0.492/0.328	0.177/**0.051**	0.242/0.090	**0.171**/0.057
Image 22	0.997/1.105	0.874/0.983	0.227/0.121	1.028/1.175	0.357/0.232	0.440/0.292	**0.214**/**0.090**
Mean	0.677/0.617	0.611/0.532	0.206/0.099	0.565/0.478	0.261/0.138	0.378/0.247	**0.195**/**0.086**
Std. dev	0.191/0.340	0.141/0.261	0.067/0.078	0.193/0.340	0.109/0.113	0.161/0.263	**0.061**/**0.067**



σ_z=0	Zhuo and Sim [21]	Shi *et al.* [37]	D'Andrès *et al.* [10]	Liu *et al.* [32]	Park *et al.* [33]	Karaali and Jung [34]	Proposed Method
Zhuo and Sim [21]	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Shi *et al.* [37]	🟥	🟥	🟥	🟥	🟥	🟥	🟥
D'Andrès *et al.* [10]	🟩	🟩	🟥	🟩	🟥	🟩	🟥
Liu *et al.* [32]	🟩	🟩	🟥	🟥	🟥	🟥	🟥
Park *et al.* [33]	🟩	🟩	🟥	🟩	🟥	🟩	🟥
Karaali and Jung [34]	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Proposed Method	🟩	🟩	🟩	🟩	🟩	🟩	🟥

(a)

σ_z=1	Zhuo and Sim [21]	Shi *et al.* [37]	D'Andrès *et al.* [10]	Liu *et al.* [32]	Park *et al.* [33]	Karaali and Jung [34]	Proposed Method
Zhuo and Sim [21]	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Shi *et al.* [37]	🟥	🟥	🟥	🟥	🟥	🟥	🟥
D'Andrès *et al.* [10]	🟩	🟩	🟥	🟩	🟩	🟩	🟥
Liu *et al.* [32]	🟩	🟥	🟥	🟥	🟥	🟥	🟥
Park *et al.* [33]	🟩	🟩	🟥	🟩	🟥	🟩	🟥
Karaali and Jung [34]	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Proposed Method	🟩	🟩	🟥	🟩	🟩	🟩	🟥

(b)

Figure 8: Significance test for DME results. (a) Clean realistic dataset. (b) Noisy realistic dataset. For each cell, if the row index method is significantly better than the column index method, it is marked as green; otherwise, it is marked as red.

can catch the defocus discontinuities reasonably well and reduce the influence of textures to a low level, as shown in Figure 9 (g). However, for strong textures, the influence cannot be eliminated perfectly (see the areas of the fence for the second image and the blue chair for the forth image). The method of [34] can catch the defocus trend and usually obtain smooth estimations (see Figure 9 (h)). However, since the defocus

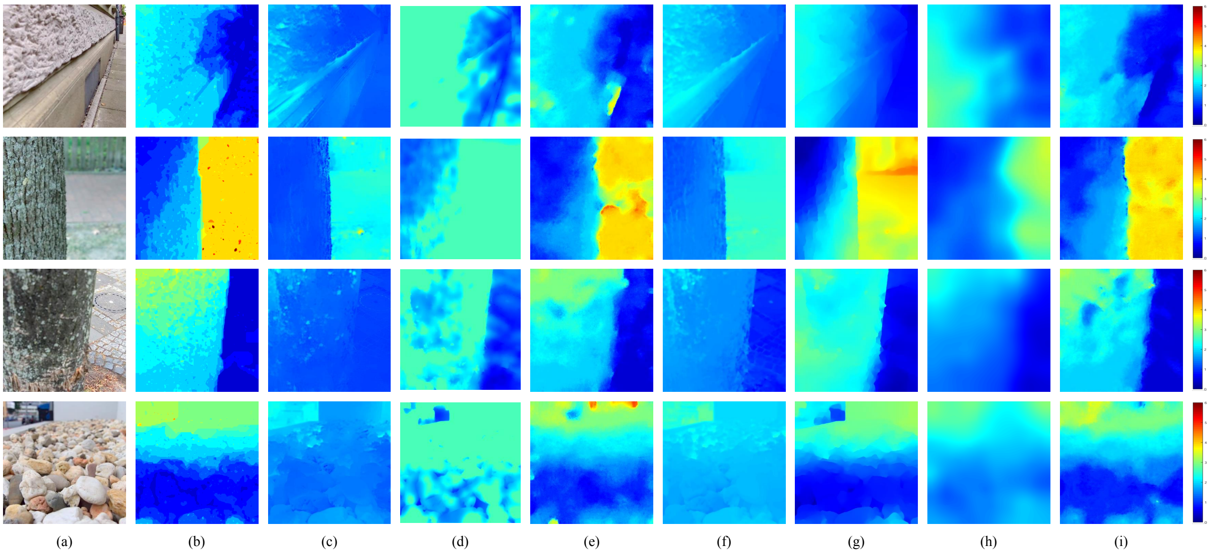


Figure 9: DME results on realistic dataset. (a) Input image. (b) Ground truth. (c) Zhuo and Sim [21]. (d) Shi *et al.* [37]. (e) D'Andrès *et al.* [10]. (f) Liu *et al.* [32]. (g) Park *et al.* [33]. (h) Karaali and Jung [34]. (i) The proposed method. Rows from top to down: image 07 without noise, image 08 without noise, image 15 with additive Gaussian white noise ($\sigma_z = 1$) and image 19 with additive Gaussian white noise ($\sigma_z = 1$).

amount is achieved by a division operation (where the numerator and denominator are the domain transform filtering results of sparse defocus map and edge map respectively), it may suffer from numerical instability. Consequently, it might obtain wrong estimations and can probably not catch the defocus discontinuities very well (see the second and third images in Figure 9 (h)). Based on region-based feature, the method of [10] can get smooth estimations for textual regions but cannot catch the defocus discontinuities very well (see the second and third images in Figure 9 (e)). The proposed method combines edge-based methods with region-based feature. Consequently, it can catch the defocus discontinuities well (see the second and third images in Figure 9 (i)), and can obtain smooth estimations for textural regions (see the areas of the wall for the first image, the tree trunk for the second and third images and the stones for the forth image).

All the methods are tested on a computer with an Intel Xeon E5 2.4GHz CPU and 128GB RAM. The average running time of each method is shown in Table 8. As can be seen, the running time of the proposed method is comparable with that of [10]. However, the error is smaller than [10], as shown in Tables 6-7 and Figure 8. Compared with [34], the proposed method is significantly superior, though its running time is about 10 times longer. Currently, the only optimization adopted in the implementation of the proposed method is parallel computing. To further improve its efficiency, there are at least two optimizations that can be done. On one hand, the algorithm for searching local maximum can be better designed, for example, coordinate descent algorithm with random initial seeds can be employed as [9] did. On the other hand, C++ implementation and code optimization can also be considered.

Table 8: Average running time on realistic dataset.

Method	Zhuo and Sim [21]	Shi *et al.* [37]	D'Andrès *et al.* [10]	Liu *et al.* [32]	Park *et al.* [33]	Karaali and Jung [34]	Proposed Method
Time (second)	14.2	9.7	162.3	49.6	34.6	1.6	184.0

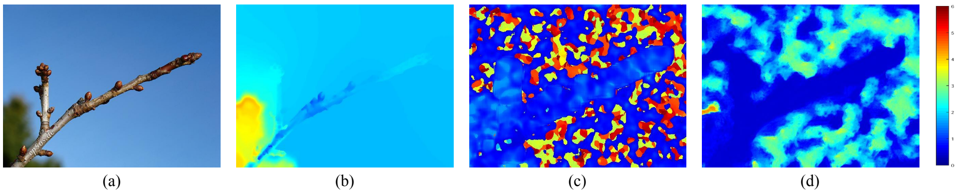


Figure 10: Example showing the limitation of the proposed method. (a) Input image. (b) Defocus map by [32], serving as one of the three linear basis. (c) The first channel of feature. (d) Estimated defocus map.

Since the method can be viewed as a unified approach of edge-based method and region-based method, it might fail when both methods produce inaccurate estimations. For example, the proposed method occasionally obtains incorrect defocus values in large homogeneous areas. For these areas, the estimations of edge-based methods (linear basis) are not reliable because they are propagated from far away edge points; the extracted feature might produce inaccurate estimations since the likelihood might be very sensitive to noise for these areas (the reason is that the response of Gabor filters will be small for homogeneous areas). See Figure 10 for an example. For an input image with large homogeneous area, both edge-based linear basis (b) and region-based feature (c) might produce inaccurate estimations. Consequently, the final result (d) is also inaccurate. Generally, this limitation is hard to overcome. To improve the results for these areas, maybe multi-scale patch size (trying to avoid homogeneous patches) can be employed in the feature extraction step as [33] did.

Another limitation of the proposed method is that it might get inaccurate estimation for areas where the defocus amount is larger than 6. This limitation comes from the training dataset since the defocus amount range of the training dataset is $[0.5, 6]$. Fortunately, it can be further addressed by designing a new training dataset with a larger range of defocus amount.

5. Applications

As mentioned in Section 1, DME can be used in many applications. In this section, we demonstrate the usage of the proposed method for two applications, i.e., defocused image deblurring and defocus blur detection. The proposed method is also compared with several competitive methods.

Table 9: Quantitative results of deblurring on clean realistic dataset.

Input	Blurry	Zhuo and	Shi *et al.*	D'Andrès *et*	Liu *et al.*	Park *et al.*	Karaali and	Propose
Image	Image	Sim [21]	[37]	*al.* [10]	[32]	[33]	Jung [34]	Method
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Image 01	25.43/0.811	22.87/0.776	21.42/0.748	**27.47/0.888**	23.25/0.794	24.46/0.824	25.06/0.845	27.41/0.886
Image 02	24.51/0.694	23.15/0.676	20.08/0.646	**25.89/0.811**	22.31/0.678	23.98/0.715	24.32/0.732	25.83/0.810
Image 03	23.99/0.875	22.56/0.873	19.51/0.808	24.28/0.907	21.17/0.857	22.91/0.889	23.08/0.888	**25.37/0.921**
Image 04	23.78/0.748	22.57/0.758	21.33/0.746	**25.46/0.842**	21.69/0.761	23.32/0.788	22.90/0.791	25.27/0.839
Image 05	25.63/0.886	22.16/0.826	20.75/0.750	**26.76/0.921**	20.38/0.781	24.23/0.875	23.12/0.859	26.69/0.917
Image 06	24.30/0.836	20.75/0.806	22.15/0.830	**25.65**/0.895	20.80/0.811	23.49/0.846	23.48/0.856	25.42/**0.897**
Image 07	27.33/0.860	24.18/0.842	22.96/0.813	27.84/0.897	23.44/0.842	25.88/0.874	25.00/0.861	**28.85/0.910**
Image 08	23.52/0.753	23.94/0.773	21.33/0.744	25.53/0.847	23.73/0.776	23.84/0.785	23.97/0.775	**25.58/0.849**
Image 09	22.81/0.764	21.16/0.689	20.04/0.690	24.29/0.852	21.64/0.734	21.73/0.740	21.79/0.748	**24.61/0.867**
Image 10	23.83/0.742	23.11/0.749	21.59/0.763	25.37/0.843	24.28/0.795	24.07/0.804	24.07/0.798	**25.46/0.847**
Image 11	23.29/0.658	22.02/0.634	21.75/0.671	24.70/0.792	22.61/0.660	23.02/0.691	22.37/0.683	**24.88/0.798**
Image 12	23.71/0.743	22.95/0.731	21.29/0.718	25.96/0.845	22.91/0.742	23.50/0.760	23.18/0.755	**26.06/0.848**
Image 13	28.08/0.871	23.11/0.813	22.32/0.785	**28.74/0.914**	22.61/0.810	26.21/0.865	26.63/0.880	27.65/0.899
Image 14	24.11/0.842	22.68/0.809	20.28/0.718	25.41/0.886	21.87/0.791	22.66/0.813	22.74/0.822	**25.61/0.896**
Image 15	20.68/0.624	19.42/0.563	19.16/0.630	22.60/0.791	19.79/0.597	19.92/0.624	20.04/0.604	**22.83/0.813**
Image 16	24.04/0.805	23.45/0.809	21.63/0.778	25.78/0.884	23.91/0.825	23.92/0.828	23.98/0.823	**25.85/0.887**
Image 17	24.43/0.606	22.10/0.579	21.24/0.612	25.51/0.760	20.64/0.585	23.67/0.622	24.28/0.627	**25.55/0.762**
Image 18	22.63/0.756	21.28/0.727	18.86/0.650	24.04/0.859	21.34/0.746	21.04/0.745	21.24/0.771	**24.28/0.862**
Image 19	24.47/0.870	22.95/0.841	20.61/0.767	26.00/0.913	21.63/0.809	23.11/0.851	22.46/0.837	**26.04/0.916**
Image 20	22.48/0.721	20.62/0.690	20.50/0.692	**24.12/0.840**	21.07/0.709	21.59/0.726	21.68/0.740	23.91/0.834
Image 21	25.53/0.923	23.99/0.892	21.18/0.842	26.07/0.930	22.87/0.876	25.48/0.920	25.39/0.921	**26.28/0.934**
Image 22	23.85/0.890	21.28/0.819	20.82/0.799	24.48/0.904	19.70/0.786	22.97/0.861	22.55/0.854	**24.71/0.909**
Mean	24.20/0.785	22.38/0.758	20.94/0.736	25.54/0.865	21.99/0.762	25.35/0.861	23.33/0.794	**25.64/0.868**
Gain	–/–	-1.82/-0.027	-3.26/-0.049	1.34/0.079	-2.22/-0.023	1.15/0.075	-0.87/0.009	**1.44/0.083**

5.1. Defocused image deblurring

One application of DME is defocused image deblurring, i.e., using the estimated defocus map to deblur the input image. The deblurring algorithm is achieved by solving the following optimization problem [34]:

$$\min_{\mathbf{x}} \|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \left(\|\mathbf{D}_{\mathrm{h}}\mathbf{x}\|_\alpha^\alpha + \|\mathbf{D}_{\mathrm{v}}\mathbf{x}\|_\alpha^\alpha \right). \tag{7}$$

Here, $\mathbf{B}$ is the blur matrix generated from the defocus map, $\mathbf{x}$ and $\mathbf{y}$ denote the deblurred and input images, $\mathbf{D}_{\mathrm{h}}$ and $\mathbf{D}_{\mathrm{v}}$ denote the horizontal and vertical discrete derivative operators, and $\mathbf{x}$ is optimized using the iteratively reweighted least squares (IRLS) method similar to [15]. Following the settings of [34], the parameters are set as: $\lambda = 2 \cdot 10^{-3}, \alpha = 0.8$.

The deblurring experiments are conducted on the realistic dataset and the quantitative results are shown in Table 9 (clean version) and Table 10 (noisy version), where the metrics are peak signal to noise ratio (PSNR) and structural similarity (SSIM), for which the higher the better. The best results are in bold. It can be found that the proposed method can obtain the best deblurring result for most images, with an average gain of 1.44dB/0.083 for clear images and 0.71dB/0.064 for noisy images. Only the methods of [10, 33] and the proposed method can produce positive gains in terms of PSNR and SSIM, meaning that only these three methods are suitable for image deblurring.

To better analyze the quantitative results, we conduct significance tests on the data of Tables 9 and 10. Since the distribution of the data might not be Gaussian distribution, a non-parametric test is adopted: we employ the Friedman test and the results are shown in Figure 11, where (a) is for the clean realistic dataset

Table 10: Quantitative results of deblurring on noisy realistic dataset ($\sigma_z = 1$).

Input Image	Blurry Image PSNR/SSIM	Zhuo and Sim [21] PSNR/SSIM	Shi *et al.* [37] PSNR/SSIM	D'Andrès *et al.* [10] PSNR/SSIM	Liu *et al.* [32] PSNR/SSIM	Park *et al.* [33] PSNR/SSIM	Karaali and Jung [34] PSNR/SSIM	Propose Method PSNR/SSIM
Image 01	25.31/0.802	22.24/0.738	22.00/0.741	25.26/0.832	23.11/0.765	24.46/0.824	24.46/0.815	**25.45**/**0.836**
Image 02	24.40/0.684	22.47/0.631	21.10/0.647	**25.02**/**0.776**	21.99/0.638	23.98/0.715	23.78/0.699	24.87/0.773
Image 03	**23.86**/0.870	21.74/0.837	20.67/0.821	22.92/0.871	20.66/0.822	22.91/**0.889**	22.45/0.860	23.50/0.870
Image 04	23.71/0.737	22.36/0.734	21.80/0.733	**24.30**/**0.801**	21.42/0.737	23.32/0.788	22.39/0.765	24.18/0.801
Image 05	**25.48**/0.868	21.69/0.764	21.55/0.739	25.22/**0.888**	20.14/0.720	24.23/0.875	22.23/0.790	25.19/0.877
Image 06	24.24/0.833	20.61/0.794	22.29/0.826	**24.98**/**0.877**	20.78/0.802	23.49/0.846	23.34/0.848	24.60/0.875
Image 07	27.27/0.857	24.10/0.835	23.24/0.813	27.08/0.884	23.57/0.836	25.88/0.874	24.84/0.854	**28.16**/**0.896**
Image 08	23.49/0.750	23.64/0.763	22.60/0.762	24.94/0.833	23.44/0.766	23.84/0.785	23.66/0.764	**25.00**/**0.836**
Image 09	22.79/0.762	21.04/0.680	20.14/0.687	23.86/0.840	21.45/0.721	21.73/0.740	21.59/0.738	**24.11**/**0.854**
Image 10	23.79/0.739	22.60/0.728	22.09/0.758	**24.36**/0.816	23.74/0.771	24.07/0.804	23.56/0.779	24.32/**0.816**
Image 11	23.24/0.651	21.67/0.612	21.99/0.670	24.50/0.781	22.44/0.639	23.02/0.691	22.14/0.667	**24.58**/**0.785**
Image 12	23.66/0.737	22.64/0.715	21.72/0.720	**25.79**/0.837	22.79/0.728	23.50/0.760	22.98/0.742	25.79/**0.837**
Image 13	**28.01**/0.869	22.85/0.801	22.73/0.788	27.84/**0.900**	22.74/0.804	26.21/0.865	26.24/0.870	27.32/0.890
Image 14	24.08/0.840	22.49/0.799	20.56/0.722	24.93/0.874	21.84/0.784	22.66/0.813	22.50/0.809	**25.08**/**0.884**
Image 15	20.65/0.621	19.23/0.546	19.32/0.608	21.87/0.752	19.67/0.581	19.92/0.624	19.88/0.589	**21.99**/**0.758**
Image 16	23.99/0.802	23.10/0.793	22.01/0.778	25.06/0.863	23.72/0.810	23.92/0.828	23.71/0.810	**25.15**/**0.867**
Image 17	24.34/0.600	21.97/0.559	21.75/0.613	25.10/**0.743**	20.60/0.559	23.67/0.622	24.01/0.609	**25.11**/0.741
Image 18	22.59/0.751	21.08/0.705	19.29/0.653	23.19/0.832	21.32/0.727	21.04/0.745	21.07/0.758	**23.32**/**0.836**
Image 19	24.43/0.866	22.80/0.830	20.91/0.766	24.97/0.894	21.54/0.800	23.11/0.851	22.21/0.825	**25.12**/**0.899**
Image 20	22.44/0.718	20.48/0.677	20.62/0.688	**23.67**/**0.825**	21.04/0.697	21.59/0.726	21.52/0.728	23.44/0.817
Image 21	25.48/0.921	23.92/0.887	21.43/0.843	25.58/0.921	22.81/0.871	25.48/0.920	25.17/0.915	**25.82**/**0.927**
Image 22	23.82/0.889	21.29/0.817	20.98/0.799	24.12/0.894	19.97/0.787	22.97/0.861	22.41/0.848	**24.55**/**0.906**
Mean	24.14/0.780	22.09/0.738	21.40/0.735	24.75/0.842	21.85/0.744	24.29/0.830	23.01/0.776	**24.85**/**0.845**
Gain	–/–	-2.05/-0.042	-2.74/-0.045	0.61/0.062	-2.29/-0.037	0.15/0.050	-1.13/-0.004	**0.71**/**0.064**

and (b) is for the noisy version. For each cell, if the row index method is significantly better than the column index method, it is marked as green; otherwise, it is marked as red. It can be found that only [10] and the proposed method can produce significantly better results than the blurry input image. These two methods are comparable.

Figure 12 gives visual results on four images from the realistic dataset (two without noise and two with noise). As can be seen, the proposed method can obtain the best deblurring results for both clear and noisy images, with the sharpest restoration (see the cirrus in the right top corner of the first image, the window and tree in the right top corner of the second image, the vertical railing in the left top corner of the third image, and the texture on the trunk of the forth image).

5.2. Defocus blur detection

Another application of DME is defocus blur detection, i.e., to detect the defocused regions in an image. Once the defocus map $\mathbf{d}$ is obtained, the defocused region can be detected by simply thresholding:

$$L_D[n] = \begin{cases} 1, & \text{if } d[n] > \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Here $L_D[n]$ is the detection result at pixel n and $\tau = (1-\beta)d_{\min} + \beta d_{\max}$ is the threshold. $d_{\min}$ and $d_{\max}$ denote the minimum and maximum of the defocus map, respectively. β is a parameter to determine the threshold. The threshold is usually determined by setting β empirically.

σ_z=0	Blurry Image	Zhuo and Sim [21]	Shi *et al.* [37]	D'Andrès *et al.* [10]	Liu *et al.* [32]	Park *et al.* [33]	Karaali and Jung [34]	Proposed Method
Blurry Image	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Zhuo and Sim [21]	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥
Shi *et al.* [37]	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
D'Andrès *et al.* [10]	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥
Liu *et al.* [32]	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟥
Park *et al.* [33]	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Karaali and Jung [34]	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Proposed Method	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥

(a)

σ_z=1	Blurry Image	Zhuo and Sim [21]	Shi *et al.* [37]	D'Andrès *et al.* [10]	Liu *et al.* [32]	Park *et al.* [33]	Karaali and Jung [34]	Proposed Method
Blurry Image	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Zhuo and Sim [21]	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Shi *et al.* [37]	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
D'Andrès *et al.* [10]	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥
Liu *et al.* [32]	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
Park *et al.* [33]	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Karaali and Jung [34]	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟥
Proposed Method	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟥

(b)

Figure 11: Significance tests for deblurring results. (a) Clean realistic dataset. (b) Noisy realistic dataset. For each cell, if the row index method is significantly better than the column index method, it is marked as green; otherwise, it is marked as red.

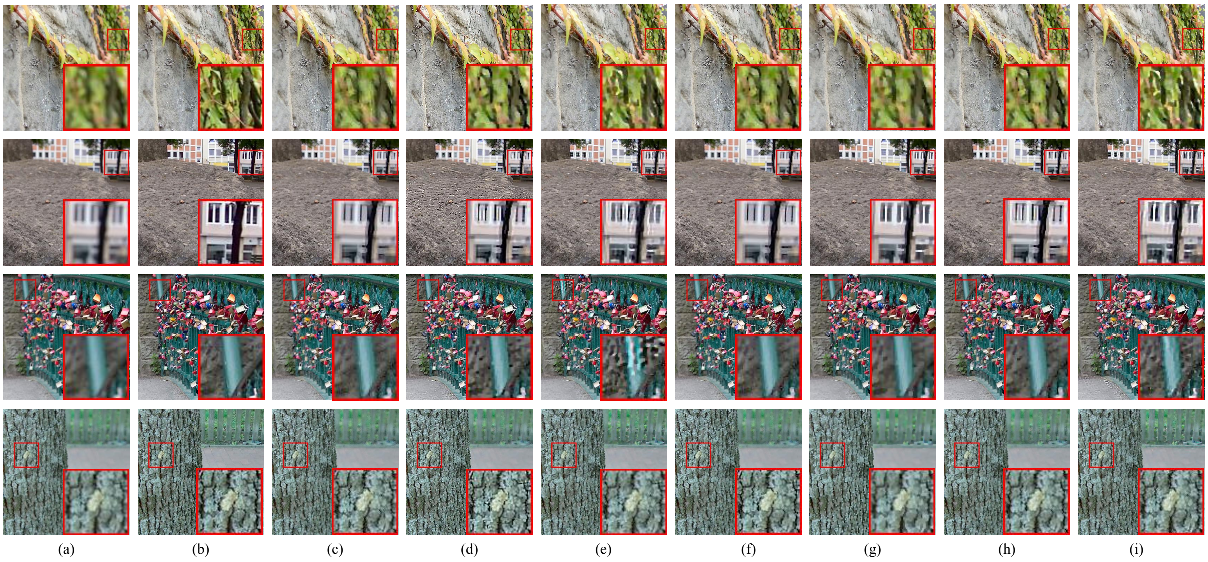


Figure 12: Deblurred results on realistic dataset. (a) Input image. (b) Ground truth all-sharp image. (c) Zhuo and Sim [21]. (d) Shi *et al.* [37]. (e) D'Andrès *et al.* [10]. (f) Liu *et al.* [32]. (g) Park *et al.* [33]. (h) Karaali and Jung [34]. (i) The proposed method. Rows from top to down: image 14 with no noise, image 18 with no noise, image 03 with additive Gaussian white noise ($\sigma_z = 1$) and image 08 with additive Gaussian white noise ($\sigma_z = 1$).

Table 11: Average accuracy of defocus detection.

Method	Zhuo and	Shi *et al.*	Shi *et al.*	D'Andrès *et*	Liu *et al.*	Park *et al.*	Karaali and	Propose
	Sim [21]	2014 [41]	2015 [37]	*al.* [10]	[32]	[33]	Jung [34]	Method
β	0.25	0.20	0.40	0.14	0.32	0.30	0.26	0.20
Average accuracy	0.757	0.728	0.711	0.735	0.758	0.796	0.755	**0.799**

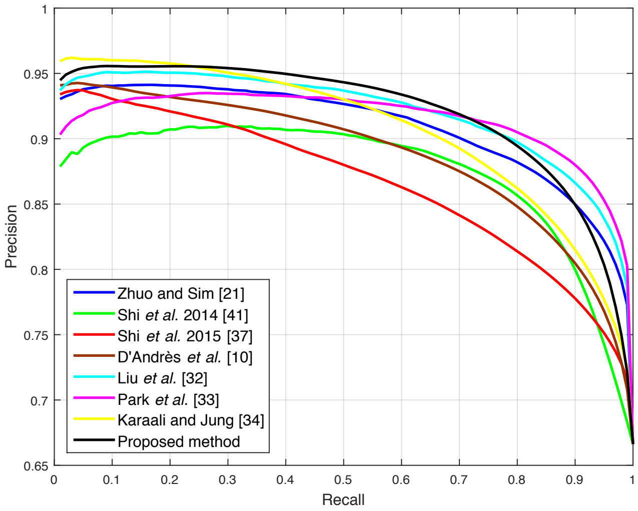


Figure 13: Average precision-recall curve.

The experiments are conducted on the CUHK dataset [41], where there are 704 partially defocused images as well as the corresponding ground truth for defocus region label. The proposed method is compared with [10, 21, 32–34, 37, 41]. The metric is detection accuracy and precision-recall curve. They are defined as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{9}$$

$$Precision = \frac{TP}{TP + FP}, \tag{10}$$

$$Recall = \frac{TP}{TP + FN}. \tag{11}$$

Here TP, TN, FP, FN are the number of pixels where the estimated label is 1 and the ground truth label is also 1, the estimated label is 0 and the ground truth label is also 0, the estimated label is 1 while the ground truth label is 0, the estimated label is 0 while the ground truth label is 1, respectively.

For different methods, β needs to be set appropriately. We set β empirically to achieve the highest average detection accuracy for each method. The exact value for β as well as the corresponding average accuracy are shown in Table 11. As can be seen, the proposed method can achieve the highest average accuracy.

By varying β from 0 to 1, we can obtain the precision-recall curve for each method. The average precision-recall curves are shown in Figure 13. It can be easily found that the proposed method and the method in

Table 12: Area under the precision-recall curve.

Method	Zhuo and Sim [21]	Shi *et al.* 2014 [41]	Shi *et al.* 2015 [37]	D'Andrès *et al.* [10]	Liu *et al.* [32]	Park *et al.* [33]	Karaali and Jung [34]	Propose Method
Area under curve	0.904	0.870	0.862	0.883	**0.915**	0.910	0.902	**0.915**

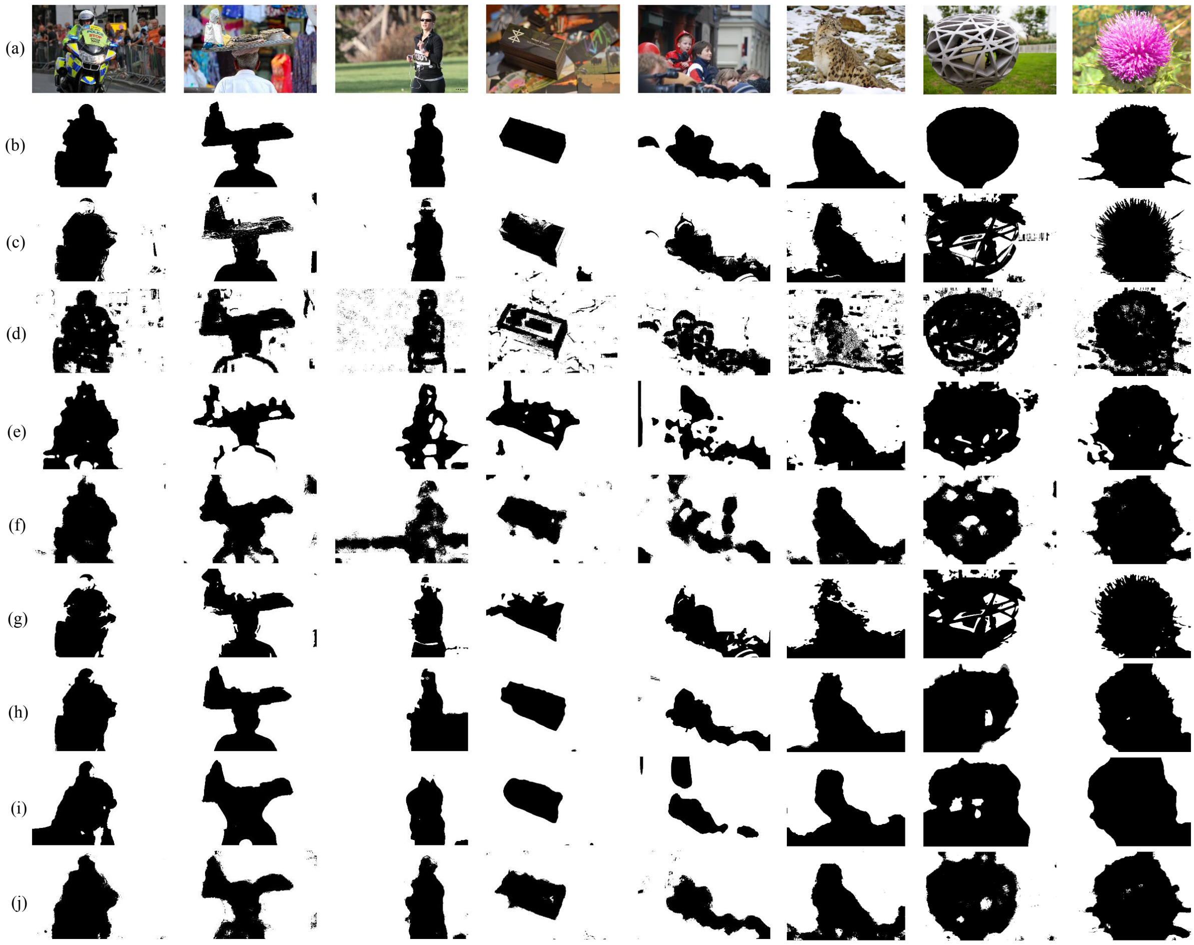


Figure 14: Visual comparison for defocus blur detection. (a) Input image. (b) Ground truth. (c) Zhuo and Sim [21]. (d) Shi *et al.* 2014 [41]. (e) Shi *et al.* 2015 [37]. (f) D'Andrès *et al.* [10]. (g) Liu *et al.* [32]. (h) Park *et al.* [33]. (i) Karaali and Jung [34]. (j) Proposed method.

[33] are dominant. Specifically, for low and middle recall, the proposed method performs the best; for high recall, the method of [33] performs the best. As pointed out in [47], the area under the precision-recall curve can reflect the average performance of a method. Therefore, the area under the curve is calculated for each method and the results are shown in Table 12. As can be seen, the proposed method and the method of [32] have the largest area, meaning that these two methods perform the best on average.

Some visual results for defocus blur detection can be found in Figure 14. As can be seen in (j), the results of the proposed method are closest to the ground truth. The method of [33] can also obtain satisfied results, as shown in (h). For [21], the defocus map is usually influenced by textures of the input image, therefore, the detection result can also be influenced by textures, as shown by the last image in (c). The method of [32] can improve the performance of [21], however, the improvement is very limited. Neither of the methods of [10] or [34] can catch the defocus discontinuities very well, therefore, their detection results cannot either catch the boundary of in-focus/defocused area, as shown in (f) and (i). The features of [41] and [37] usually cannot distinguish high level defocused area and smooth area, therefore, there are often holes in the in-focus area, as shown in (d) and (e).

6. Conclusion

In this paper, we propose a single image DME method based on localized 2D frequency analysis and RTF. Two orthogonal gradient operators with the corresponding Gabor filter subsets are employed to calculate the likelihood, avoiding the negative impact of very small filter responses. After the likelihoods are calculated, only the first K highest local maximums are taken as input feature of the RTF, reducing the feature dimension without losing its power of expressiveness. Furthermore, the superiority of edge-based methods are incorporated in our method by designing the linear basis of RTF as the results of three edge-based methods. These improvements enable the proposed method to get better defocus maps. The experiments demonstrate that our method outperforms both edge-based and region-based DME methods. Finally, the proposed method is successfully applied to defocused image deblurring and defocus blur detection. One limitation of the proposed method is that it might occasionally obtain incorrect defocus values in large homogeneous areas where both the input feature and the linear basis are inaccurately estimated. Since the lack of RGB-defocus amount datasets is the bottleneck for developing end-to-end deep learning based DME methods, in the future, we plan to first break through this bottleneck by building a much larger realistic dataset than that used in this paper, and then try to explore end-to-end deep learning based methods for the DME problem.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant 61771276, the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen under Grant JCYJ20170817161056260, the Initial Funding of Newly-Introduced Teacher in Shenzhen University with No. 2019121, and the Tsinghua Scholarship for Overseas Graduate Studies with No. 2017041. The authors would like to thank all the editors and reviewers for their useful comments and suggestions on our manuscript. The authors would also like to thank Mr. Laurent D'Andrès (the first author of [10]) for sharing the synthetic and realistic dataset.

References

References

[1] V. P. Namboodiri, S. Chaudhuri, On defocus, diffusion and depth estimation, Pattern Recognition Letters 28 (3) (2007) 311–319. doi:10.1016/j.patrec.2006.04.011.

[2] H. Sheng, P. Zhao, S. Zhang, J. Zhang, D. Yang, Occlusion-aware depth estimation for light field using multi-orientation EPIs, Pattern Recognition 74 (2018) 587–599. doi:10.1016/j.patcog.2017.09.010.

[3] S. Bae, F. Durand, Defocus magnification, Computer Graphics Forum 26 (3) (2007) 571–579. doi:10.1111/j.1467-8659.2007.01080.x.

[4] J. Gu, G. Meng, S. Xiang, C. Pan, Blind image quality assessment via learnable attention-based pooling, Pattern Recognition 91 (2019) 332–344. doi:10.1016/j.patcog.2019.02.021.

[5] W. Zhang, W.-K. Cham, Single-image refocusing and defocusing, IEEE Transactions on Image Processing 21 (2) (2012) 873–882. doi:10.1109/TIP.2011.2162739.

[6] S.-S. Tung, W.-L. Hwang, Multiple depth layers and all-in-focus image generations by blurring and deblurring operations, Pattern Recognition 69 (2017) 184–198. doi:10.1016/j.patcog.2017.03.035.

[7] Z. Chen, J. Yuan, Y. P. Tan, Hybrid saliency detection for images, IEEE Signal Processing Letters 20 (1) (2013) 95–98. doi:10.1109/LSP.2012.2230442.

[8] Z. Wang, Y. Ma, J. Gu, Multi-focus image fusion using PCNN, Pattern Recognition 43 (6) (2010) 2003–2016. doi:10.1016/j.patcog.2010.01.011.

[9] X. Zhu, S. Cohen, S. Schiller, P. Milanfar, Estimating spatially varying defocus blur from a single image, IEEE Transactions on Image Processing 22 (12) (2013) 4879–4891. doi:10.1109/TIP.2013.2279316.

[10] L. D'Andrès, J. Salvador, A. Kochale, S. Süsstrunk, Non-parametric blur map regression for depth of field extension, IEEE Transactions on Image Processing 25 (4) (2016) 1660–1673. doi:10.1109/TIP.2016.2526907.

[11] F. Deschênes, D. Ziou, P. Fuchs, Improved estimation of defocus blur and spatial shifts in spatial domain: a homotopy-based approach, Pattern Recognition 36 (9) (2003) 2105–2125. doi:10.1016/S0031-3203(03)00040-2.

[12] F. Deschênes, D. Ziou, P. Fuchs, A homotopy-based approach for computing defocus blur and affine transform simultaneously, Pattern Recognition 41 (7) (2008) 2263–2282. doi:10.1016/j.patcog.2007.12.005.

[13] H. Liu, Y. Jia, H. Cheng, S. Wei, Depth recovery from defocus images using total variation, in: 2010 Second International Conference on Computer Modeling and Simulation, Vol. 2, 2010, pp. 146–150. doi:10.1109/ICCMS.2010.261.

[14] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, P. Hanrahan, Light field photography with a hand-held plenoptic camera, Computer Science Technical Report CSTR 2 (11) (2005) 1–11.

[15] A. Levin, R. Fergus, F. Durand, W. T. Freeman, Image and depth from a conventional camera with a coded aperture, ACM Transactions on Graphics 26 (3) (2007) 70. doi:10.1145/1276377.1276464.

[16] F. Moreno-Noguer, P. N. Belhumeur, S. K. Nayar, Active refocusing of images and videos, ACM Transactions on Graphics 24 (2) (2007) 67. doi:10.1145/1239451.1239518.

[17] J. Jancsary, S. Nowozin, T. Sharp, C. Rother, Regression tree fields—an efficient, non-parametric approach to image labeling problems, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2376–2383. doi:10.1109/CVPR.2012.6247950.

[18] A. Chakrabarti, T. Zickler, W. T. Freeman, Analyzing spatially-varying blur, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2512–2519. doi:10.1109/CVPR.2010.5539954.

[19] Y.-W. Tai, M. S. Brown, Single image defocus map estimation using local contrast prior, in: 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 1797–1800. doi:10.1109/ICIP.2009.5414620.

[20] H. Cheong, E. Chae, E. Lee, G. Jo, J. Paik, Fast image restoration for spatially varying defocus blur of imaging sensor, Sensors 15 (1) (2015) 880–898. doi:10.3390/s150100880.

[21] S. Zhuo, T. Sim, Defocus map estimation from a single image, Pattern Recognition 44 (9) (2011) 1852–1858. doi:10.1016/j.patcog.2011.03.009.

[22] A. Levin, D. Lischinski, Y. Weiss, A closed-form solution to natural image matting, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2) (2008) 228–242. doi:10.1109/TPAMI.2007.1177.

[23] Y. Cao, S. Fang, Z. Wang, Digital multi-focusing from a single photograph taken with an uncalibrated conventional camera, IEEE Transactions on Image Processing 22 (9) (2013) 3703–3714. doi:10.1109/TIP.2013.2270086.

[24] K. He, J. Sun, X. Tang, Guided image filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (6) (2013) 1397–1409. doi:10.1109/TPAMI.2012.213.

[25] C. Tang, C. Hou, Z. Song, Defocus map estimation from a single image via spectrum contrast, Optics Letters 38 (10) (2013) 1706–1708. doi:10.1364/OL.38.001706.

[26] A. Karaali, C. R. Jung, Adaptive scale selection for multiresolution defocus blur estimation, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 4597–4601. doi:10.1109/ICIP.2014.7025932.

[27] A. Nasonov, A. Nasonova, A. Krylov, Edge width estimation for defocus map from a single image, in: Advanced Concepts for Intelligent Vision Systems, 2015, pp. 15–22. doi:10.1007/978-3-319-25903-1_2.

[28] G. Xu, Y. Quan, H. Ji, Estimating defocus blur via rank of local patches, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5381–5389. doi:10.1109/ICCV.2017.574.

[29] X. Zhang, R. Wang, X. Jiang, W. Wang, W. Gao, Spatially variant defocus blur map estimation and deblurring from a single image, Journal of Visual Communication and Image Representation 35 (2016) 257–264. doi:10.1016/j.jvcir.2016.01.002.

[30] Q. Chen, D. Li, C. K. Tang, KNN matting, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (9) (2013) 2175–2188. doi:10.1109/TPAMI.2013.18.

[31] D. J. Chen, H. T. Chen, L. W. Chang, Fast defocus map estimation, in: 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 3962–3966. doi:10.1109/ICIP.2016.7533103.

[32] S. Liu, F. Zhou, Q. Liao, Defocus map estimation from a single image based on two-parameter defocus model, IEEE Transactions on Image Processing 25 (12) (2016) 5943–5956. doi:10.1109/TIP.2016.2617460.

[33] J. Park, Y. Tai, D. Cho, I. S. Kweon, A unified approach of multi-scale deep and hand-crafted features for defocus estimation, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2760–2769. doi:10.1109/CVPR.2017.295.

[34] A. Karaali, C. R. Jung, Edge-based defocus blur estimation with adaptive scale selection, IEEE Transactions on Image Processing 27 (3) (2018) 1126–1137. doi:10.1109/TIP.2017.2771563.

[35] J. Oliveira, M. Figueiredo, J. Bioucas-Dias, Parametric blur estimation for blind restoration of natural images: Linear motion and out-of-focus, IEEE Transactions on Image Processing 23 (1) (2014) 466–477. doi:10.1109/TIP.2013.2286328.

[36] R. Yan, L. Shao, Blind image blur estimation via deep learning, IEEE Transactions on Image Processing 25 (4) (2016) 1910–1921. doi:10.1109/TIP.2016.2535273.

[37] J. Shi, L. Xu, J. Jia, Just noticeable defocus blur detection and estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 657–665. doi:10.1109/CVPR.2015.7298665.

[38] C. Tang, J. Wu, Y. Hou, P. Wang, W. Li, A spectral and spatial approach of coarse-to-fine blurred image region detection, IEEE Signal Processing Letters 23 (11) (2016) 1652–1656. doi:10.1109/LSP.2016.2611608.

[39] S. A. Golestaneh, L. J. Karam, Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 596–605. doi:10.1109/CVPR.2017.71.

[40] H. Xiao, W. Lu, R. Li, N. Zhong, Y. Yeung, J. Chen, F. Xue, W. Sun, Defocus blur detection based on multiscale SVD fusion in gradient domain, Journal of Visual Communication and Image Representation 59 (2019) 52–61. doi:10.1016/j.jvcir.2018.12.048.

[41] J. Shi, L. Xu, J. Jia, Discriminative blur detection features, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2965–2972. doi:10.1109/CVPR.2014.379.

[42] W. Zhao, F. Zhao, D. Wang, H. Lu, Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3080–3088. doi:10.1109/CVPR.2018.00325.

[43] S. Zhang, X. Shen, Z. Lin, R. Mech, J. P. Costeira, J. M. F. Moura, Learning to understand image blur, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 6586–6595. doi:10.1109/CVPR.2018.00689.

[44] K. Zeng, Y. Wang, J. Mao, J. Liu, W. Peng, N. Chen, A local metric for defocus blur detection based on CNN feature learning, IEEE Transactions on Image Processing 28 (5) (2019) 2107–2115. doi:10.1109/TIP.2018.2881830.

[45] J. Jancsary, S. Nowozin, C. Rother, Loss-specific training of non-parametric image restoration models: A new state of the art, in: Computer Vision – ECCV 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 112–125. doi:10.1007/978-3-642-33786-4_9.

[46] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 898–916. doi:10.1109/TPAMI.2010.161.

[47] Z. Sun, F. Zhou, Q. Liao, A robust feature descriptor based on multiple gradient-related features, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 1408–1412. doi:10.1109/ICASSP.2017.7952388.