Estimating Defocus Blur via Rank of Local Patches

Guodong Xu[1], Yuhui Quan[2], Hui Ji[1]
[1]Department of Mathematics, National University of Singapore, Singapore 119076
[2]School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
{a0109978@u.nus.edu, csyhquan@scut.edu.cn, matjh@nus.edu.sg}

Abstract

This paper addresses the problem of defocus map estimation from a single image. We present a fast yet effective approach to estimate the spatially varying amounts of defocus blur at edge locations, which is based on the maximum ranks of the corresponding local patches with different orientations in gradient domain. Such an approach is motivated by the theoretical analysis which reveals the connection between the rank of a local patch blurred by a defocus-blur kernel and the blur amount by the kernel. After the amounts of defocus blur at edge locations are obtained, a complete defocus map is generated by a standard propagation procedure. The proposed method is extensively evaluated on real image datasets, and the experimental results show its superior performance to existing approaches.

1. Introduction

Conventional cameras produce images with best sharpness when the objects of a scene are exactly on the focal plane of focusing module. The further is an object away from the focal plane, the more blurred it appears in the image, as shown in Fig. 2 (a). Such a phenomenon is called *defocus (or out-of-focus)* whose blur amount is related to the translation of the object away from the focal plane along optical axis, as illustrated in Fig. 1. More specifically, when an object is placed at the focal distance d_f, all light beams from any point of the object will converge to a single sensor point, which leads to image pixels with best sharpness. In contrast, the light beams from the points with the distance $d \neq d_f$ will arrive at a region with multiple sensor points, which leads to blurred image pixels. Such a region is called circle of confusion (CoC).

Defocus amount and scene depth. The *defocus amount* of a pixel, denoted by c, is defined as the diameter of CoC ([8]). The defocus amount c is related to the *scene depth*, denoted by d, as follows:

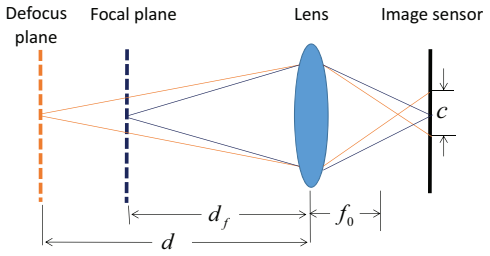


Figure 1: Illustration of focus and defocus [37].

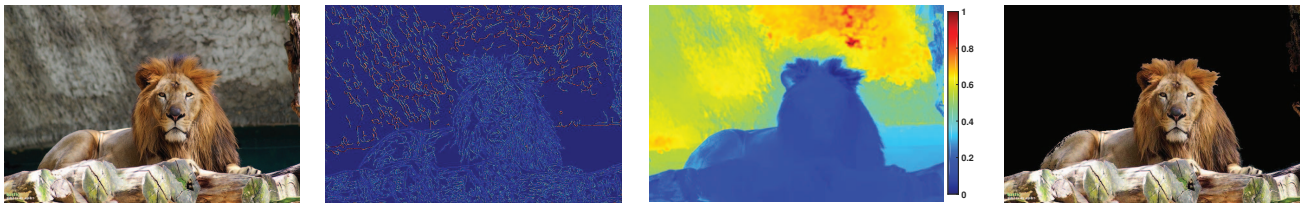
$$c = \frac{|d - d_f|}{d} \frac{f_0^2}{n_s(d_f - f_0)}, \tag{1}$$

where n_s is the stop number and f_0 is the focal length. Clearly, the defocus amount c monotonically increases when the scene depth f increases. Thus, for an image I captured for the scene with varying depth, the defocus amount is spatially varying. We define the *defocus map* of an image as the matrix c whose (i, j)-th entry $c[i, j]$ is the defocus amount of the pixel at $[i, j]$.

Defocus amount and blur kernel. Defocus map is also closely related to the image degradation caused by out-of-focus, as it measures the blur amount of each pixel of an out-of-focus image. For example, as the blurring effect is often modeled as local averaging weighted by 2D isotropic Gaussian functions, local regions of defocused image can then be modeled by the convolution between sharp image regions and isotropic Gaussian kernels with spatially varying standard deviation (s.t.d.), denoted by $\sigma[i, j]$. The s.t.d. $\sigma[i, j]$ is equivalent to the defocus map $c[i, j]$ up to a constant, i.e. $\sigma[i, j] = \kappa_0 c[i, j]$ for some global constant κ_0. See *e.g.* [7, 9, 25] for more details. In other words, defocus map is equivalent to the s.t.d. of spatially varying blur kernels of an out-of-focus image.

Applications. Since defocus map provides essential in-

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.574



(a) (b) (c) (d)

Figure 2: Demonstration of defocus map estimation and its application to in-focus/defocused region segmentation. (a) Input; (b) Defocus amount estimation on edge points (normalized to $[0, 1]$) by the proposed method; (c) Complete defocus map estimation (normalized to $[0, 1]$); (d)In-focus regions detected using the defocus map.

formation of image degradation caused by out-of-focus, it has been used in many applications in image processing and computational photography; *e.g*. image quality assessment [31], image deblurring [14, 27], image refocusing [33, 34] and defocus magnification [2, 34].

Moreover, for an image pixel, Equation (1) provides a closed-form relationship between its defocus amount and its scene depth. It can be seen that the defocus amount monotonically increases when scene depth increases, as long it is larger than the focus distance. In other words, the defocus map of an image can be used as the ordinal depth map of a scene. Ordinal depth map can see its wide applications in computer vision and computer graphics. For example, defocus map has been used for depth estimation [5, 26], image segmentation [29, 12] and image matting [15]. See Fig. 2 for an illustration of defocus map estimation and the resulting segmentation of in-focus regions.

1.1. Related work

Based on how many input images are available, existing approaches for defocus estimation can be classified into two categories: multi-image approach (*e.g*. [22, 23, 35]) and single-image approach (*e.g*. [37, 34, 30, 20]). The multi-image approaches employ auxiliary equipments (*e.g*. coded aperture cameras [35]), or set different camera parameters for generating multiple images of the same scene, and the defocus amount is estimated by triangulation. The applicability of these multi-image methods is limited, since they work well only for the static scenes where multiple images are well aligned and no occlusions exist. As this paper is about estimating the defocus map from a single input image, we do not give a detailed review on the multi-image methods but rather focus on the discussion of the single-image methods.

The single-image approaches attempt to estimate the defocus map from a single input image. One effective scheme is to introduce extra active capture processes for defocus map estimation by either manipulating illumination conditions [16] or introducing new camera devices [10]. In recent years, many single-image methods have been proposed, which do not require additional capture processes and hence can be used for commodity cameras; see *e.g*. [2, 37, 34, 30, 20]. As smooth image regions contain little information of blurriness, most of these single-image methods take a two-stage scheme, *i.e*., a sparse defocus map is first computed by only estimating defocus amount along image edges, and then the full defocus map is constructed by propagating the available defocus amount estimation to all image pixels.

Regarding the defocus amount estimation on image edges, Elder and Zucker [6] modeled defocus around an edge as a convolution of a step function with a Gaussian kernel. The s.t.d. of the Gaussian kernel is used for measuring defocus amount, and it is estimated from the distance between the second derivative extrema of opposite sign in the gradient direction. Using the same model as [6], Zhuo and Sim [37] proposed to estimate the blur amount of edge pixels using the ratio of gradient magnitudes between the input image and a re-blurred image convoluted by a Gaussian kernel. This method produces very impressive results on some images. However, it cannot handle image edges well when two or more edges are very close [37], as re-blurring will merge these image edges. Recently, Shi [28] proposed a method based on the sparse representation over a dictionary learned from a set of images with different contents. Similar concept with pre-defined dictionary (edgelet) was also proposed in [26], which estimates the blur amount on a small piece of edge by matching the edge with an edgelet set. As these two methods were designed to estimate small blur amount (i.e. the so-called just noticeable blur in [28]), they are not very suitable for processing the images with significant defocus blur.

Once a sparse defocus map along image edges is obtained, several methods have been proposed in the past to generate a full defocus map; see *e.g*. [2, 37]. Bae *et al*. [2] extends the work of [6] by using an inverse diffusion method to interpolate a full defocus map from the sparse one, as well as using bilateral filtering to remove the outliers in the estimates. Zhuo and Sim [37] proposed to use the matting Laplacian method [11] for propagating the

sparse map, which empirically yields better visual results than the inverse diffusion method used in [2].

Another alternative single-image approach is to exploit the frequency information of image edges for defocus estimation; see *e.g*. [30, 4, 36]. Tang *et al*. [30] utilizes spectrum contrast to estimate the defocus amount at edge locations. In [4], sub-band decomposition is combined with Gaussian scale mixtures for estimating the likelihood function of a given candidate blur kernel. This method is extended to the continuous domain in [36], and the extension also incorporates many other processes, including localized spectrum analysis, color edge detection and smoothness constraints.

1.2. Main idea and contributions

In this paper, we first proposed an effective metric for defocus amount estimation at edge points. By viewing a defocused local patch (matrix) as an in-focus patch convoluted by an out-of-focus kernel, e.g. an isotropic 2D Gaussian kernel. Our mathematical analysis reveals that the matrix rank of a patch will decrease when the patch is blurred by a Gaussian kernel, and the matrix rank monotonically decreases when the s.t.d. of the Gaussian kernel increases. Moreover, if the in-focus patch satisfies certain properties, *e.g*. positive (negative) definiteness, the s.t.d. of the Gaussian kernel can be directly estimated from the matrix rank of the patch. These results lead to the introduction of a new rank-based metric for defocus amount on edges.

Secondly, to exploit the rank-based metric for defocus amount, we developed a construction scheme of local patches in image gradient domain for estimating the defocus amount on image edges. The construction is based on two observations on the patches of an in-focus image in gradient domain: (1) the local gradient patches centered at edge points are usually of narrow band with dominant values of the same sign; and (2) a rank-deficient band matrix is very likely to be strictly diagonally dominant after being rotated by 45 degree or 135 degree. In other words, if we symmetrically sample the gradient patches which are centered at an edge point with different orientations, at least one of these sampled patches is very likely to be positive (negative) definite. Then, the maximum rank (deficient rank) of such multi-oriented patches of a defocused edge point will reveal the s.t.d. of the corresponding Gaussian kernel, i.e. its associated defocus amount.

The proposed approach has several advantages over existing single-image methods in terms of robustness and accuracy.

- Compared to [37, 2], the proposed method does not require image edges are well separated and thus can effectively process texture regions.
- Compared to [30], the proposed method does not require the in-focus region has a dense distribution of image edges than the out-of-focus region.
- Compared to [26, 28] which focus on images with just noticeable defocus blur, the proposed method can effectively process images with significant defocus blur.

These advantages of the proposed method over others are also justified by extensive experiments on real data.

2. Rank-based metric of defocus amount

We first introduce some notations. Throughout this paper, the indexes of vectors and matrices start with 0. For a vector $g \in \mathbb{R}^n$, let $g[j]$ denote the $(j+1)$-th element of g, $\|g\|_0$ denote the ℓ_0-pseudo-norm of g that counts the number of non-zero entries in g, and $\widehat{g} \in \mathbb{C}^n$ denote its discrete Fourier transform (DFT). For a matrix $G \in \mathbb{R}^{n_1 \times n_2}$, let $G[i,j]$ denote the $(i+1, j+1)$-th entry of G and $\text{rank}(G)$ denote the rank of G. For any $X, Y \in \mathbb{R}^{n_1 \times n_2}$, let $X \circledast Y$ denote the discrete convolution between X and Y.

A defocused image patch can be viewed as the convolution between an in-focus image patch and an out-of-focus kernel. The same relationship also holds for the patches generated in image gradient domain. Through this paper, we define patches in image gradient domain. In the next, we establish the rank-based relationship between a gradient patch and its defocused version. Let $U \in \mathbb{R}^{n \times n}$ denote a in-focus patch, $I \in \mathbb{R}^{n \times n}$ denote the defocus blurred version of U, and $G \in \mathbb{R}^{n \times n}$ denote the associated convolution kernel. For simplicity, we assume G is symmetric [1]. It is known in linear algebra that a symmetric matrix can be decomposed into the summation of rank-one matrices:

$$G = \sum_{i=1}^{\text{rank}(G)} \lambda_i g_i g_i^\top \tag{2}$$

where g_is (λ_is) are the eigenvectors (eigenvalues) of G.

Proposition 1. *Consider three matrices $U, I, G \in \mathbb{R}^{n \times n}$ related by $I = G \circledast U$. Let $G = \sum_{i=1}^{rank(G)} \lambda_i g_i g_i^T$. Then,*

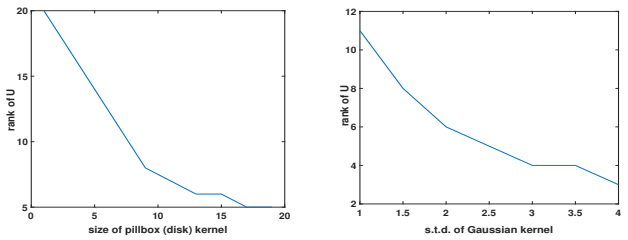
$$rank(I) \leq \sum_{i=1}^{rank(G)} \|\widehat{g}_i\|_0. \tag{3}$$

Proof. See Appendix A for the detailed proof. □

The isotropic 2D Gaussian kernel arguably is the most often-seen out-of-focus kernel; see e.g. [2, 8, 37]. An isotropic 2D Gaussian filter can be expressed as $G = gg^\top$ for some 1D Gaussian filter $g \in \mathbb{R}^n$. By Proposition 1, the rank of the defocused patch is less than $\|\widehat{g}\|_0$. Furthermore, it is known that the Fourier transform of a Gaussian with s.t.d. σ is still a Gaussian with s.t.d. $\frac{1}{\sigma}$. For a 2D Gaussian kernel $G = gg^\top$, we have that $\|\widehat{g}\|_0$[2]. monotonically de-

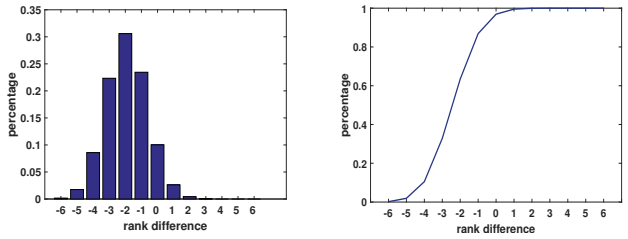
[1] Most defocus kernels, *e.g*. Gaussian and pillbox, are symmetric.

[2] The implementation of $\|\cdot\|_0$ treats the values less than 10^{-2} as zero.



(a) (b)

Figure 3: Illustration of the relation between rank(U) and (a) the size of pillbox k or (b) the s.t.d. σ of Gaussian kernel .



(a) (b)

Figure 4: (a) Normalized histogram of the number of gradient patches over the rank differences of before- and after-defocus blurring; and (b) its cumulative curve.

creases when the s.t.d. σ of the Gaussian kernel increases. Thus, as long as the rank of the in-focus patch is higher than $\|\widehat{g}\|_0$, the rank of the defocused patch will decrease. Another often-seen out-of-focus kernel is the pillbox (disk) filter; see e.g. [26, 32]. It also has the similar behavior.

See Fig. 3 for an illustration of Proposition 1, where I is a 20×20 identity matrix, and G is the convolution matrix w.r.t. a pillbox kernel or a Gaussian kernel. It can be seen that the rank of $U = I \otimes G$ decreases when the size k of the pillbox kernel or the s.t.d. σ of the Gaussian kernel increases.

The statement of Proposition 1 is also consistent with the empirical experiments on real images. From the Multi-focus Image Dataset [17], we randomly sampled 5×10^4 gradient patches with size 9×9 at edge points and their defocused correspondences. Then, the rank difference between each pair (U, I), i.e. $\text{rank}(U) - \text{rank}(I)$, is calculated. See Fig. 4 for the normalized histogram regarding the number of image patches versus the rank difference. It can be seen that around 90% in-focus image patches have their ranks decreased after being blurred by defocus. This clearly indicates the validity of the statement on the rank decreasing of defocused patches in Proposition 1.

Now, suppose we can construct in-focus patch U which is a positive (negative) definite matrix, then we can establish the formula which relates $\|\widehat{g}\|_0$ to the rank of the defocused patch I. Taking Gaussian kernel for example, we have the following proposition:

Proposition 2. *Consider three matrices $U, I, G \in \mathbb{R}^{n\times n}$ related by $I = G \circledast U$. Suppose that the matrix U is positive (negative) definite, and $G = gg^T$. Then, we have*

$$\textit{rank}(I) = \|\widehat{g}\|_0. \tag{4}$$

Proof. See Appendix B for the detailed proof. □

Notice that $\|\widehat{g}\|_0$ is solely determined by the parameter of the out-of-focus kernel. Proposition 2 implies that the rank of defocused patch I can be used for estimating the defocus amount (*e.g.* the s.t.d. σ of a Gaussian kernel G), as long as we can construct the patches that have the same convolution relationship and are positive (negative) definite.

3. Defocus map estimation

In this section, we develop a scheme of constructing suitable patches that are applicable to Proposition 2 and thus can be used for estimating defocus amount. The key idea is that a rank deficient matrix can become of full rank by rotation, which can be done by sampling patches from the input image with different orientations. In other words, for each edge point, we sample gradient patches from the input image with different orientations to ensure that there exists at least one with full rank among these patches.

As smooth image regions contain little information regarding defocus, we first only estimate the defocus amount on image edge points, which leads to a sparse defocus map. Such a process is done in image gradient domain. Given a single gray-scale image I, we first detect all edge points using some edge detector, e.g., the Canny edge detector [3]. It is noted that the output of an edge detector is used here only for getting the set of edge points. The edges themselves are not used for estimating defocus amount. The gradient of I on these edge points are then calculated by some gradient operator ∇.[3]

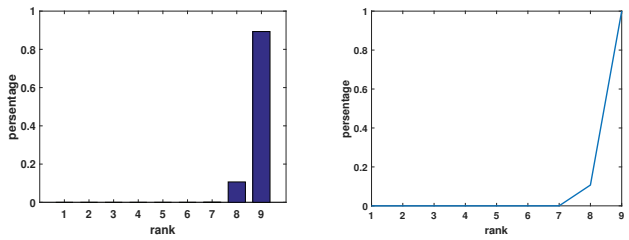
For each edge point (i_0, j_0), we sample totally four $(2p+1) \times (2p+1)$ patches from ∇I along different orientations, that is

$$\begin{cases} Q_0[i,j] = \nabla I[i_0 - p + i, j_0 - p + j]: & \text{horizontal;} \\ Q_1[i,j] = \nabla I[i_0 - p + j, j_0 + p - i]: & \text{vertical;} \\ Q_2[i,j] = \nabla I[i_0 + i - j, j_0 - p + i]: & \text{diagonal;} \\ Q_3[i,j] = \nabla I[i_0 - p + i, j_0 - i + j]: & \text{anti-diagonal.} \end{cases} \tag{5}$$

Then we define four symmetric matrices $\{P_k\}_{k=0}^3$ by:

$$P_k = Q_k + Q_k^\top, \quad 0 \le k \le 3. \tag{6}$$

[3]The implementation uses $\nabla = \frac{\partial}{\partial_x} + \frac{\partial}{\partial_y}$, where partial derivatives are calculated by the filter $[1, -1]$.



(a) (b)

Figure 5: (a) Normalized histogram of the number of gradient patches over the maximum rank of oriented patches in in-focus regions; and (b) its cumulative curve.

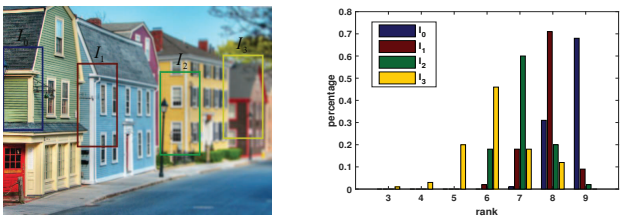
For a local image region centering at an edge point, its gradient content usually contains either single or multiple image edges with different orientations. In the case of single image edge, the associated matrix is roughly of narrow band. It will be rank-deficient only when the orientation of image edge is far away from diagonal or anti-diagonal. The four patches with different sampling orientations essentially guarantee that there exists at least one whose corresponding image edge is close to diagonal or anti-diagonal, i.e., one of $\{Q_k\}_{k=0}^3$ will be of full rank. In the case of multiple image edges with different orientations, the matrix is more likely to be of full rank, since it is the summation of multiple matrices associated with single image edge. The treatment of symmetry, $\{P_k\}_{k=0}^3$, is likely to even further increase the rank of a rank-deficient matrix. Such an assertion is also consistent with the statistics done on Multi-focus Image Dataset [17]. See Fig. 5 for the histogram regarding the maximum rank of four patches of size 9×9 randomly selected from 10^6 edge points within in-focus regions of all images in the dataset[4].

By Proposition 2, together with the fact that all these four patches can be viewed as being blurred by the same out-of-focus kernel, we have that the defocus blur amount can be determined by the value of $\max_{0\le k\le 3} \text{rank}(P_k)$. By numerical simulation, we propose the following formula:

$$c^{-1} \sim -\ln(1 - \max_{0\le k\le 3} \text{rank}(P_k)/n), \qquad (7)$$

where n is patch size. Such an estimation formula is demonstrated in Fig. 6. The sample defocused image in Fig. 6 is mainly composed of four regions: one in-focus region and three defocused regions with noticeably different defocus amounts, as marked out by four rectangles. See Fig. 6b for the normalized histogram regarding the number of edge points versus the maximum rank of the corresponding patches. It can be seen that for most edge points in in-focus regions, the maximum rank of the constructed patches is full; for the edge points in defocused regions, the maximum rank of the constructed patches is of lower value.

[4]The rank function was implemented by treating singular value values less than 10^{-3} as zero.



(a) (b)

Figure 6: Distribution of maximum ranks of patches of edge points with different defocus amount. (a) Sample image and fours regions with different defocus amount; (b) Normalized histogram of the number of edge points from the four selected regions over the corresponding maximum ranks.

In the previous step, we only estimate the defocus amounts of edge points detected by Canny edge detector. The obtained defocus map is then sparse. To reconstruct the full defocus map, we follow other two-stage defocus map estimation methods, *e.g.*, Zhuo and Sim [37], to propagate the available defocus amount at edges to the whole image by the matting Laplacian method [11]. The propagation is done by keeping the resulting defocus amount close to the given ones at edge points, and meanwhile keeping the discontinuities of defocus map consistent with that of image edges. Interested readers can refer to [11] for more details. As the defocus estimations on edge locations might be occasionally erroneous, same as [2, 37], we also use bilateral filtering [19] to pre-process the sparse defocus map before being inputed to the matting Laplacian method. The whole algorithm for defocus map estimation is summarized Alg. 1.

Algorithm 1 Defocus map estimation

1: **INPUT:** Defocused image I.
2: **OUTPUT:** Defocus map σ.
3: Calculating gradient image ∇I.
4: Constructing a set of edge points using Canny edge detector.
5: **for** each edge point **do**
6: Constructing four oriented symmetric patches by (5) and (6).
7: Calculating the maximum rank of four patches.
8: **end for**
9: Constructing a sparse defocus map by (7) using the maximum rank of patches on each edge point.
10: Reconstructing the full defocus map σ by the matting Laplacian method [11].

Petunia Bottle Forest Flower Farm Bird Dog Withy

Figure 7: Examples of test images.

4. Experiments

In this section, the proposed method is evaluated on the real images which were collected from the defocus image dataset [37] and the RetargetMe dataset [24], as well as from on-line resources. See Fig. 7 for the samples from the tested dataset. The dataset covers most often-seen defocus scenarios, *e.g*. the image "Bottle" whose in-focus region contains both cartoon regions and textures and whose defocused regions have four depth layers, the image "Forest" whose both in-focus and defocused regions contain many small edges, the image "Flower" whose the in-focus regions have less edges than the defocused regions.

The proposed method is compared to five other recent defocus map estimation methods with code available online, including Bae *et al*. 's method [2] (Bae), Zhuo *et al*. 's method [37] (Tang), Tang *et al*. 's method [30], Shi *et al*.'s method [28] (Shi-I) and Shi *et al*.'s method [26] (Shi-II). All of these methods have codes published online. The results of these compared methods on the test datasets are directly cited from the literature if possible and re-produced otherwise using published codes with rigorous parameter tuning.

The parameters of the proposed method are set the same for all test images. In the estimation of defocus amount on edge points, the patch size is set to 9×9, and the rank function is implemented by treating singular values less than 10^{-1} as zero. In the completion of full defocus map, the implementation of the matting Laplacian method [11] is exactly the one used in [37] with the default parameter setting (i.e. the key parameter $\lambda = 10^{-3}$)[5]. See Fig. 9 for an illustration of the results from the two stages.

4.1. Visual comparison of defocus map

Fig. 9 showed the results of defocus amount estimation for image "Petunia" on edge points from the proposed one and Zhuo's *et al*.'s methods for comparison. While both used the same map completion method in the second stage, the outcomes are rather different owing to the different results in the first stage. Zhuo *et al*. 's method produced erroneous estimation on the in-focus region, i.e. the petunia contains blur texture as indicated in white rectangle which is not blurred by defocus, and such errors worsen the accuracy of estimated defocus map. In contrast, our method produced a more accurate depth map which indicts a more accurate estimation of defocus amounts on edge points.

[5] http://github.com/phervo/ProjetRD48/tree/master/Sources/Matlab/DefocusEstimation_Sources

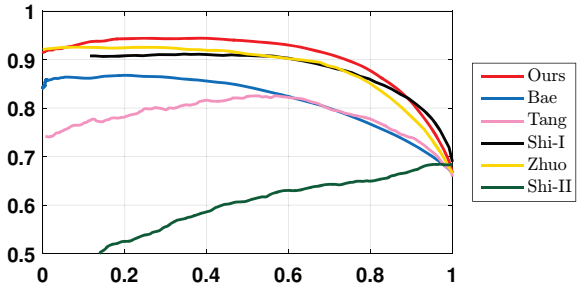


Figure 8: Precision and recall curves of in-focus/defocused region segmentation using the defocus maps generated by different methods.

Methods	Bae	Zhuo	Tang	Shi-I	Shi-II	Ours
F_β-measure	0.78	0.84	0.78	0.84	0.75	**0.86**

Table 1: The largest F_β-measure each method got.

See Fig. 10 for the comparison of the results from different methods on three sample images. More results can be found in the supplementary materials. It can be seen that the proposed method noticeably outperformed the other compared methods in terms of the accuracy of ordinal depth and the boundaries of ordinal depths. In comparison, the performance of Bae *et al*.'s method and Tang *et al*.'s method in general is not satisfactory. Zhuo *et al*.'s method is overall the second best, but it did not handle texture regions well. For example, it produced erroneous estimation on in-focus region of the image "Bottle" and the image "Forest" with a very dense distribution of image edges. Regarding Shi-I and Shi-II, both methods claimed that they are specifically designed for stimating small (just noticeable) defocus blur. Such a claim is also consistent with experimental observations, as they did not perform well on images with large defocus blur, e.g. the images "Bottle" and "Flower".

4.2. Quantitative Evaluation

The most often-seen application of defocus map estimation is in-focus/defocused region segmentation, as defocus map provides an estimation of defocus amount. See Fig. 9 for illustration. Same as [28], we quantitatively evaluated the accuracy of defocus map, using the quality measurement of in-focus/defocused region segmentation on the tested defocus image dataset[6]. The dataset contains (i) 704 images with defocused regions and (ii) truth of segmentation map which are manually done. The defocused regions of the images from this dataset have a wide range of blur degrees from small HLblur to large blur.

Each image is segmented into a sharp region and

[6] http://www.cse.cuhk.edu.hk/~leojia/projects/dblurdetect/dataset.html

a blurred region by applying simple thresholding (with threshold value T_{seg}) to the defocus map. The segmentation results then are evaluated via precision and recall:

$$\text{precision} = |R \cap R_g|/|R|, \;\; \text{recall} = |R \cap R_g|/|R_g|,$$

where R, R_g denote the pixel sets corresponding to the segmented blurred region and the ground truth, and $|\cdot|$ denotes the size of the set. The precision and recall curves of all methods for comparison were generated with respect to different thresholds T_{seg}. See Fig. 8 for the precision and recall curves of all methods. It can be seen that most of the time, our method has the highest precision among all methods. Also, in terms of the F_β-measure [21] (β^2 was set to be 0.3 as in [18, 1]), which is defined as the weighted harmonic mean of precision and recall, the proposed method outperforms all other methods. See Table 1 for the details. This indicates the advantage of the proposed method over others when being used for in-focus/defocused region segmentation.

5. Conclusion

This paper presented a simple yet effective method for estimating the defocus map from a single input image. Motived by the theoretical analysis that reveals the connection between the ranks of local patches and the defocus amount, we developed a rank-based metric for estimating the defocus amount along edge points by constructing specific local patches at edge points. A full defocus map is then reconstructed by a standard procedure. The proposed defocus map estimation method has many advantages over the existing ones, and extensive experiments also show the proposed one noticeably outperforms those related works.

A. Proof of Proposition 1

Proof. By eigenvalue decomposition, we have

$$G = \sum_{i=1}^{\text{rank}(G)} \lambda_i G_i$$

where $G_i = g_i {g_i}^\top$ and g_is (λ_is) are the eigenvectors (eigenvalues) of G. Then, by the linearity of convolution operator,

$$\text{rank}(G \circledast U) = \text{rank}\Big(\sum_{i=1}^{\text{rank}(G)} \lambda_i (G_i \circledast U)\Big).$$

Moreover, by the definition of 2D circular convolution, we have for $0 \le x, y < n$,

$$\begin{aligned} & G_i \circledast U[x,y] \\ =& \sum_{p=0}^{n-1}\sum_{q=0}^{n-1} U[p,q] G_i[(x-p) \bmod n, (y-q) \bmod n] \\ =& \sum_{q=0}^{n-1} \Big(\sum_{p=0}^{n-1} U[x,y] g_i[(x-p) \bmod n] \Big) g_i[(y-q) \bmod n]. \end{aligned}$$

In other words, the convolution by a rank-one matrix can be viewed as running a 1D convolution along columns (rows) followed by running another 1D convolution along rows (columns). Such a convolution can be expressed in the matrix form:

$$G_i \circledast U = \tilde{G}_i U \tilde{G}_i^*,$$

where $\tilde{G}_i \in \mathbb{R}^{n\times n}$ is the circulant matrix defined by $\tilde{G}_i[p,q] = g_i[(p-q) \bmod n]$, for $i = 0, 1, \ldots, m$. It is known that

$$\mathcal{F}^* \tilde{G}_i \mathcal{F} = \Sigma(\widehat{g}_i),$$

where $\Sigma(\widehat{g}_i)$ denote the diagonal matrix with $k+1$-th diagonal entry $\widehat{g}_i[k]$. Then, by the fact that $\mathcal{F}$ is unitary, we have

$$\text{rank}\Big(\sum_{i=1}^{\text{rank}(G)} \lambda_i G_i \circledast U \Big) = \text{rank}\Big(\sum_{i=1}^{\text{rank}(G)} \lambda_i \Sigma(\widehat{g}_i) \mathcal{F} U \mathcal{F}^* \Sigma(\widehat{g}_i^*) \Big).$$

By standard rank inequality (see *e.g.* [13]), and the fact that F is unitary, we have

$$\begin{aligned} & \text{rank}\Big(\sum_{i=1}^{\text{rank}(G)} \lambda_i \Sigma(\widehat{g}_i) \mathcal{F} U \mathcal{F}^* \Sigma(\widehat{g}_i^*) \Big) \\ \le & \sum_{i=1}^{\text{rank}(G)} \text{rank}\big(\lambda_i \Sigma(\widehat{g}_i) \mathcal{F} U \mathcal{F}^* \Sigma(\widehat{g}_i^*) \big) \\ \le & \sum_{i=1}^{\text{rank}(G)} \min\big(\text{rank}(\Sigma(\widehat{g}_i)), \text{rank}(\Sigma(\widehat{g}_i^*)), \text{rank}(U) \big). \end{aligned}$$

Together with $\text{rank}(\Sigma(\widehat{g}_i)) = \text{rank}(\Sigma(\widehat{g}_i^*)) = \|\widehat{g}_i\|_0$, we have (3). □

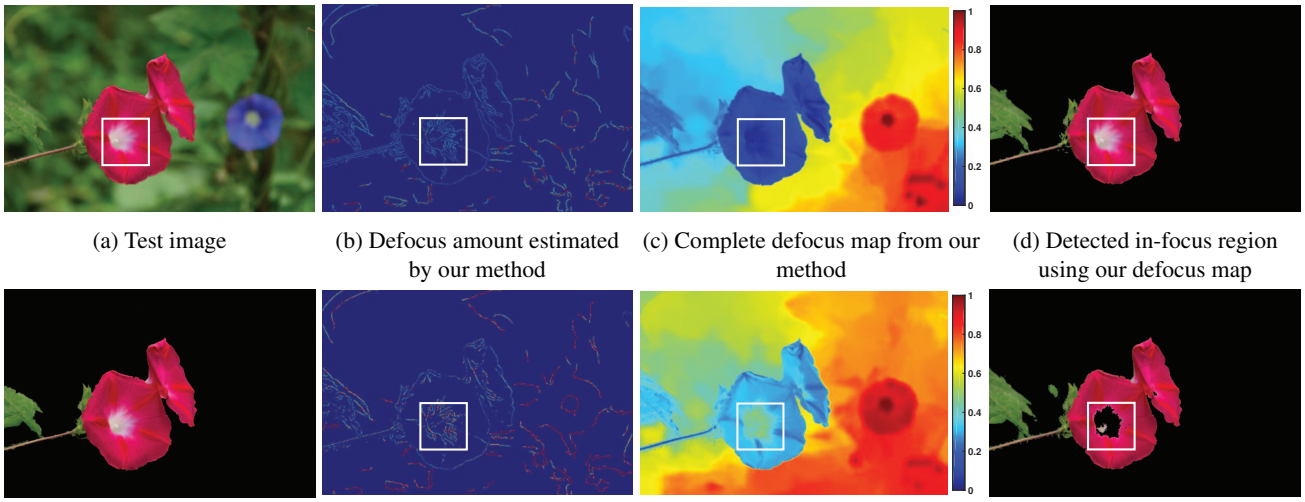
B. Proof of Proposition 2

Proof. Using the same arguments as the proof of Proposition 1, we have

$$\text{rank}(G \circledast U) = \text{rank}(\Sigma(\widehat{g})(FUF^*)\Sigma(\widehat{g}^*)).$$

As $\mathcal{F}$ is unitary and U is positive (negative)-definite, the matrix $\mathcal{F}U\mathcal{F}^*$ is also positive (negative) definite. Define $r = \|\widehat{g}\|_0$. Without loss of generality, expressing $\Sigma(\widehat{g})$ by

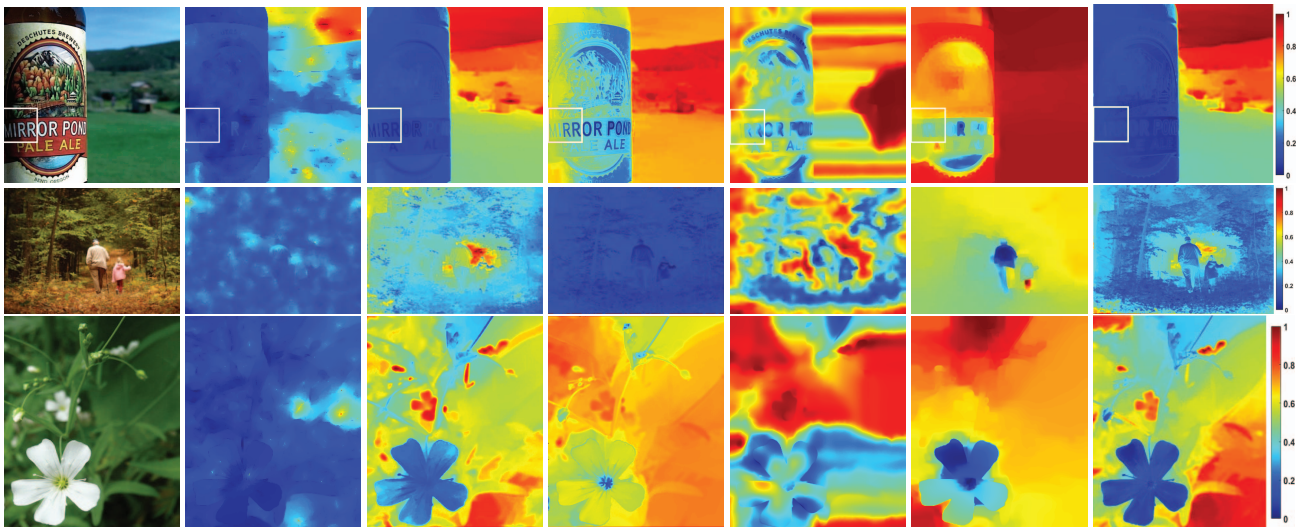
$$\Sigma(\widehat{g}) = \begin{bmatrix} \Sigma_{\widehat{g}\neq 0} & 0 \\ 0 & 0 \end{bmatrix}, \quad F_n U F_n^* = \begin{bmatrix} A & B \\ B^* & C \end{bmatrix},$$



(a) Test image (b) Defocus amount estimated by our method (c) Complete defocus map from our method (d) Detected in-focus region using our defocus map

(e) Ground truth of in-focus regions (f) Defocus amount estimated by Zhuo *et al.*'s method (g) Complete defocus map from Zhuo *et al.*'s method (h) Detected in-focus region using Zhuo *et al.*'s defocus map

Figure 9: Defocus map estimation experiment using a real image. Defocus map is normalized to $[0, 1]$. Our detected in-focus region achieves largest F_β-measure 0.988, while Zhuo *et al.*'s achieves 0.966.



(a) Input image (b) Bae [2] (c) Zhuo [37] (d) Tang [30] (e) Shi-I [28] (f) Shi-II [26] (g) Ours

Figure 10: Comparison of defocus map estimated by the tested methods.

where $\Sigma_{\widehat{g}\neq 0}$ is the $r \times r$ principal sub-matrix whose diagonal entries are the non-zero entries of $\widehat{g}$, and thus is non-singular. Then

$$\Sigma(\widehat{g})FUF^*\Sigma(\widehat{g}^*) = \begin{bmatrix} \Sigma_{\widehat{g}\neq 0} A \Sigma_{\widehat{g}^*\neq 0} & 0 \\ 0 & 0 \end{bmatrix},$$

Recall that a principal sub-matrix of a positive (negative) definite matrix is also positive (negative) definite. Thus, both $\Sigma_{\widehat{g}\neq 0}$ and A are $r \times r$ non-singular matrices. So, we have $\text{rank}(G \circledast U) = \text{rank}(\Sigma_{\widehat{g}\neq 0} A \Sigma_{\widehat{g}^*\neq 0}) = r$. The proof is done. □

Acknowledgment. This work was partially supported by Singapore MOE AcRF Research Grant MOE2012-T3-1-008 and R-146-000-229-114. Yuhui Quan would like to thank the support by National Natural Science Foundation of China (Grand No. 61602184) and Science and Technology Program of Guangzhou (Grand No. 201707010147).

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604. IEEE, 2009.

[2] S. Bae and F. Durand. Defocus magnification. In *CGF*, volume 26, pages 571–579. Wiley Online Library, 2007.

[3] J. Canny. A computational approach to edge detection. *TPAMI*, (6):679–698, 1986.

[4] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *CVPR*, pages 2512–2519. IEEE, 2010.

[5] P. P. Chan, B.-Z. Jing, W. W. Ng, and D. S. Yeung. Depth estimation from a single image using defocus cues. In *ICMLC*, volume 4, pages 1732–1738. IEEE, 2011.

[6] J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *TPAMI*, 20(7):699–716, 1998.

[7] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *TPAMI*, 27(3):406–417, 2005.

[8] E. Hecht. Optics, 4th. *Addison-Wesley*, 3, 2002.

[9] A. Kubota and K. Aizawa. Reconstructing arbitrarily focused images from two differently focused images using linear filters. *TIP*, 14(11):1848–1859, 2005.

[10] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *TOG*, 26(3):70, 2007.

[11] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *TPAMI*, 30(2):228–242, 2008.

[12] R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *CVPR*, pages 1–8, 2008.

[13] G. Marchuk. *Methods of numerical mathematics*.

[14] B. Masia, A. Corrales, L. Presa, and D. Gutierrez. Coded apertures for defocus deblurring. In *SICG*, 2011.

[15] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. In *TOG*, volume 24, pages 567–576. ACM, 2005.

[16] F. Moreno-Noguer, P. N. Belhumeur, and S. K. Nayar. Active refocusing of images and videos. *TOG*, 26(3):67, 2007.

[17] M. Nejati, S. Samavi, and S. Shirani. Multi-focus image fusion using dictionary-based sparse representation. *Information Fusion*, 25:72–84, 2015.

[18] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740. IEEE, 2012.

[19] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *TOG*, 23(3):664–672, 2004.

[20] F. Pi, Y. Zhang, G. Lu, and B. Pang. Defocus blur estimation from multi-scale gradients. In *ICMV*, pages 878309–878309. International Society for Optics and Photonics, 2013.

[21] D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[22] A. Rajagopalan and S. Chaudhuri. Maximum likelihood estimation of blur from multiple observations. In *ICASSP*, pages 2577–2580, 1997.

[23] A. Rajagopalan and S. Chaudhuri. Optimal selection of camera parameters for recovery of depth from defocused images. In *CSCVPR*, pages 219–224, 1997.

[24] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir. A comparative study of image retargeting. In *TOG*, volume 29, page 160. ACM, 2010.

[25] T. Sakamoto. Model for spherical aberration in a single radial gradient-rod lens. *Applied Optics*, 23(11):1707–1710, 1984.

[26] J. Shi, X. Tao, L. Xu, and J. Jia. Break ames room illusion: depth from general single images. *TOG*, 34(6):225, 2015.

[27] J. Shi, L. Xu, and J. Jia. Discriminative blur detection features. In *CVPR*, pages 2965–2972, 2014.

[28] J. Shi, L. Xu, and J. Jia. Just noticeable defocus blur detection and estimation. In *CVPR*, pages 657–665. IEEE, 2015.

[29] C. Swain and T. Chen. Defocus-based image segmentation. In *ICASS*, volume 4, pages 2403–2406. IEEE, 1995.

[30] C. Tang, C. Hou, and Z. Song. Defocus map estimation from a single image via spectrum contrast. *Optics Letters*, 38(10):1706–1708, 2013.

[31] X. Wang, B. Tian, C. Liang, and D. Shi. Blind image quality assessment for measuring image blur. In *CISP*, volume 1, pages 467–470. IEEE, 2008.

[32] M. Watanabe and S. K. Nayar. Rational filters for passive depth from defocus. *IJCV*, 27(3):203–225, 1998.

[33] W. Zhang and W.-K. Cham. Single image focus editing. In *ICCV*, pages 1947–1954. IEEE, 2009.

[34] W. Zhang and W.-K. Cham. Single-image refocusing and defocusing. *TIP*, 21(2):873–882, 2012.

[35] C. Zhou, O. Cossairt, and S. Nayar. Depth from diffusion. In *CVPR*, pages 1110–1117. IEEE, 2010.

[36] X. Zhu, S. Cohen, S. Schiller, and P. Milanfar. Estimating spatially varying defocus blur from a single image. *TIP*, 22(12):4879–4891, 2013.

[37] S. Zhuo and T. Sim. Defocus map estimation from a single image. *PR*, 44(9):1852–1858, 2011.