Estimating Defocus Blur via Rank of Local Patches

Guodong Xu\textsuperscript{1}, Yuhui Quan\textsuperscript{2}, Hui Ji\textsuperscript{1}
\textsuperscript{1}Department of Mathematics, National University of Singapore, Singapore 119076
\textsuperscript{2}School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
{a0109978@u.nus.edu, cayhuan@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:

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

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 $\kappa_0$. 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-
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 degrees or 135 degrees. 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 \( 0 \)-pseudo-norm of \( g \) that counts the number of non-zero entries in \( g \), and \( \hat{g} \in \mathbb{C}^n \) 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 \( X \odot Y \) 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 an 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. It is known in linear algebra that a symmetric matrix can be decomposed into the summation of rank-one matrices:

\[
G = \sum_{i=1}^{\text{rank}(G)} \lambda_i g_i g_i^\top
\]  

where \( g_i \) (\( \lambda_i \)) are the eigenvectors (eigenvalues) of \( G \).

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

\[
\text{rank}(I) \leq \sum_{i=1}^{\text{rank}(G)} \|g_i\|_0.
\]

**Proof.** See Appendix A for the detailed proof. \( \square \)

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 \( \|g\|_0 \). Furthermore, it is known that the Fourier transform of a Gaussian with s.t.d. \( \sigma \) is still a Gaussian with s.t.d. \( \frac{\sigma}{\sqrt{2}} \). For a 2D Gaussian kernel \( G = gg^\top \), we have that \( \|g\|_0 \), monotonically de-

\[\text{Most defocus kernels, e.g. Gaussian and pillbox, are symmetric.}\]

\[\text{The implementation of } \| \cdot \|_0 \text{ treats the values less than } 10^{-2} \text{ as zero.}\]
 increases 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 \( \|g\|_0 \), 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 × 10⁴ 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 \( \|\tilde{g}\|_0 \) 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 \( I = G \odot U \). Suppose that the matrix \( U \) is positive (negative) definite, and \( G = gg^T \). Then, we have

\[
\text{rank}(I) = \|\tilde{g}\|_0.
\]

**Proof.** See Appendix B for the detailed proof.

Notice that \( \|\tilde{g}\|_0 \) 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 \( \nabla \).

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

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

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

\[
P_k = Q_k + Q_k^T, \quad 0 \leq k \leq 3.
\]

The implementation uses \( \nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \), where partial derivatives are calculated by the filter \([1, -1]\).
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 \( \max_{0 \leq k \leq 3} \text{rank}(P_k) \). By numerical simulation, we propose the following formula:

\[
c^{-1} \sim -\ln(1 - \max_{0 \leq k \leq 3} \text{rank}(P_k)/n), \tag{7}
\]

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.

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.

\textbf{Algorithm 1 Defocus map estimation}

1. \textbf{INPUT:} Defocused image \( I \).
2. \textbf{OUTPUT:} Defocus map \( \sigma \).
3. Calculating gradient image \( \nabla I \).
4. Constructing a set of edge points using Canny edge detector.
5. \textbf{for} each edge point \textbf{do}
6. Constructing four oriented symmetric patches by (5) and (6).
7. Calculating the maximum rank of four patches.
8. \textbf{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 \( \sigma \) by the matting Laplacian method [11].

\textbf{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.

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

\footnote{The rank function was implemented by treating singular value values less than 10\(^{-3}\) as zero.}
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 $9 \times 9$, 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 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 in 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 indeed a more accurate estimation of defocus amounts on edge points.

$^5$http://github.com/phervo/ProjetRD48/tree/master/Sources/Matlab/DefocusEstimation_Sources

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 test-defocused image dataset$^6$. The dataset contains (i) 704 images with defocused regions and (ii) truth of segmentation 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/blurdetect/dataset.html

![Figure 7: Examples of test images.](image7.png)

![Figure 8: Precision and recall curves of in-focus/defocused region segmentation using the defocus maps generated by different methods.](image8.png)

![Figure 9: 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.](image9.png)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Bae</th>
<th>Zhuo</th>
<th>Tang</th>
<th>Shi-I</th>
<th>Shi-II</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_\beta$-measure</td>
<td>0.78</td>
<td>0.84</td>
<td>0.78</td>
<td>0.84</td>
<td>0.75</td>
<td><strong>0.86</strong></td>
</tr>
</tbody>
</table>

Table 1: The largest $F_\beta$-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 stimulating 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”.
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:

$$\text{precision} = \frac{|R \cap R_g|}{|R|}, \quad \text{recall} = \frac{|R \cap R_g|}{|R_g|},$$

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-measure [21] ($\beta^2$ 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 those related works. Moreover, by the definition of 2D circular convolution, we have for $0 \leq x, y < n$,

$$G_i \odot U[x, y] = \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} U[p, q] G_i[(x - p) \mod n, (y - q) \mod n]$$

$$= \sum_{q=0}^{n-1} \left( \sum_{p=0}^{n-1} U[x, y] g_i[(x - p) \mod n] \right) g_i[(y - q) \mod n].$$

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:

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

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

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

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

$$\text{rank}(\sum_{i=1}^{\text{rank}(G)} \lambda_i G_i \odot U) = \text{rank}(\sum_{i=1}^{\text{rank}(G)} \lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))),$$

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

$$\text{rank}(\sum_{i=1}^{\text{rank}(G)} \lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))) \leq \sum_{i=1}^{\text{rank}(G)} \text{rank}(\lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))),$$

$$\leq \sum_{i=1}^{\text{rank}(G)} \text{min}(\text{rank}(\Sigma(\tilde{g}_i)), \text{rank}(\Sigma(\tilde{g}_i^*)), \text{rank}(U)).$$

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

\section{A. Proof of Proposition 1}

\textbf{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 $g_i$ ($\lambda_i$ is) are the eigenvectors (eigenvalues) of $G$. Then, by the linearity of convolution operator,

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

\section{B. Proof of Proposition 2}

\textbf{Proof.} Using the same arguments as the proof of Proposition 1, we have

$$\text{rank}(G \odot U) = \text{rank}(\Sigma(\tilde{g}))(FUF^* \Sigma(\tilde{g}^*))$$

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

$$\Sigma(\tilde{g}) = \left[ \begin{array}{cc} \Sigma_{\tilde{g} \neq 0} & 0 \\ 0 & 0 \end{array} \right], \quad F_n U F_n^* = \left[ \begin{array}{cc} A & B \\ B^* & C \end{array} \right].$$

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

$$G_i \odot U[x, y] = \sum_{p=0}^{n-1} \sum_{q=0}^{n-1} U[p, q] G_i[(x - p) \mod n, (y - q) \mod n]$$

$$= \sum_{q=0}^{n-1} \left( \sum_{p=0}^{n-1} U[x, y] g_i[(x - p) \mod n] \right) g_i[(y - q) \mod n].$$

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:

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

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

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

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

$$\text{rank}(\sum_{i=1}^{\text{rank}(G)} \lambda_i G_i \odot U) = \text{rank}(\sum_{i=1}^{\text{rank}(G)} \lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))),$$

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

$$\text{rank}(\sum_{i=1}^{\text{rank}(G)} \lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))) \leq \sum_{i=1}^{\text{rank}(G)} \text{rank}(\lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))),$$

$$\leq \sum_{i=1}^{\text{rank}(G)} \text{rank}(\lambda_i(\Sigma(\tilde{g}_i)) F U F^* \Sigma(\tilde{g}_i^*))),$$

$$\leq \sum_{i=1}^{\text{rank}(G)} \text{min}(\text{rank}(\Sigma(\tilde{g}_i)), \text{rank}(\Sigma(\tilde{g}_i^*)), \text{rank}(U)).$$

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

\section{A. Proof of Proposition 1}

\textbf{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 $g_i$ ($\lambda_i$ is) are the eigenvectors (eigenvalues) of $G$. Then, by the linearity of convolution operator,

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

\section{B. Proof of Proposition 2}

\textbf{Proof.} Using the same arguments as the proof of Proposition 1, we have

$$\text{rank}(G \odot U) = \text{rank}(\Sigma(\tilde{g}))(FUF^* \Sigma(\tilde{g}^*))$$

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

$$\Sigma(\tilde{g}) = \left[ \begin{array}{cc} \Sigma_{\tilde{g} \neq 0} & 0 \\ 0 & 0 \end{array} \right], \quad F_n U F_n^* = \left[ \begin{array}{cc} A & B \\ B^* & C \end{array} \right].$$
Figure 9: Defocus map estimation experiment using a real image. Defocus map is normalized to $[0, 1]$. Our detected in-focus region achieves largest $F_\beta$-measure 0.988, while Zhuo et al.’s achieves 0.966.

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

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

Recall that a principal sub-matrix of a positive (negative) definite matrix is also positive (negative) definite. Thus, both $\Sigma_{\hat{g} \neq 0}$ and $A$ are $r \times r$ non-singular matrices. So, we have $\text{rank}(G \otimes U) = \text{rank}(\Sigma_{\hat{g} \neq 0}A\Sigma_{\hat{g} \neq 0}) = r$. 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 (Grant No. 61602184) and Science and Technology Program of Guangzhou (Grant No. 201707010147).
References


Supplementary Materials of "Estimating Defocus Blur via Rank of Local Patches"

Guodong Xu¹, Yuhui Quan², Hui Ji¹
¹Department of Mathematics, National University of Singapore, Singapore 119076
²School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
{a0109978@u.nus.edu, csyhquan@scut.edu.cn, matjh@nus.edu.sg}
1. Full defocus map estimation of additional test images by several test methods
Figure 2: Defocus map estimation of additional real images by several test methods, the defocus map is normalized to $[0, 1]$. 
References