Defocus Map Estimation from a Single Image

Shaojie Zhuo Terence Sim

School of Computing, National University of Singapore, Computing 1, 13 Computing Drive, Singapore 117417, SINGAPOUR

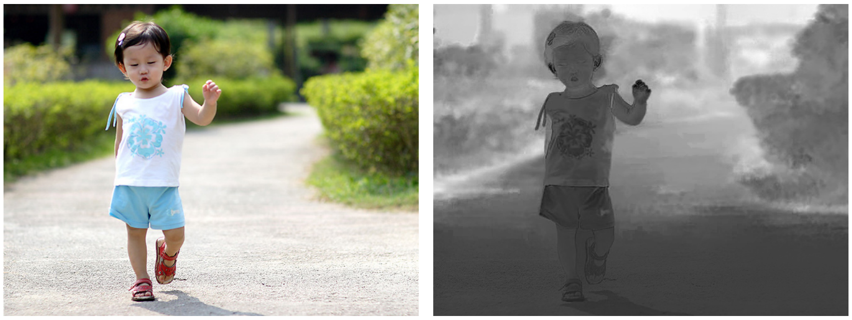
Abstract

In this paper, we address the challenging problem of recovering the defocus map from a single image. We present a simple yet effective approach to estimate the amount of spatially varying defocus blur at edge locations. The input defocused image is re-blurred using a Gaussian kernel and the defocus blur amount can be obtained from the ratio between the gradients of input and re-blurred images. By propagating the blur amount at edge locations to the entire image, a full defocus map can be obtained. Experimental results on synthetic and real images demonstrate the effectiveness of our method in providing a reliable estimation of the defocus map.

Keywords: Image processing, defocus map, defocus blur, Gaussian gradient, defocus magnification

1. Introduction

Defocus estimation plays an important role in many computer vision and computer graphics applications including depth estimation, image quality assessment, image deblurring and refocusing. Conventional methods for defocus estimation have relied on multiple images [1, 2, 3, 4]. A set of images of the same scene are captured using multiple focus settings. Then the defocus



Input

Defocus map

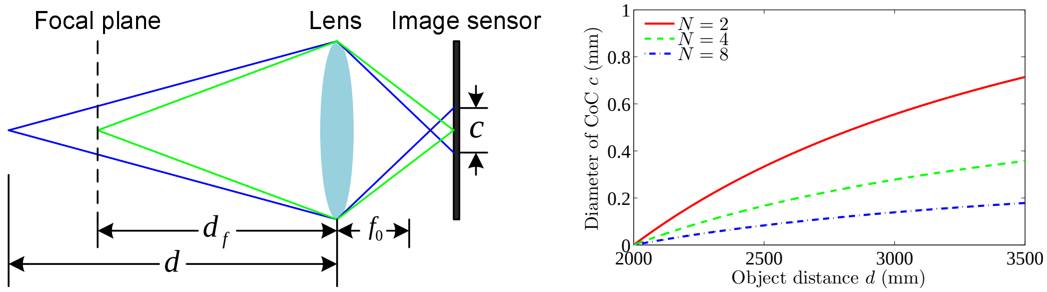
Figure 1: The depth recovery result of our method. The larger intensity means larger depth in all depth maps presented in this paper.

is measured during a implicit or explicit deblurring process. Recently, image pairs captured using coded aperture cameras [5] are used for better defocus blur measure and all-focused image recovery. However, these methods suffer from the occlusion problem and require the scene to be static, which limits their applications in practice.

In very specific settings, several methods have been proposed to recover defocus map from a single image. Active illumination methods [6] project sparse grid dots onto the scene and the defocus blur of those dots is measured by comparing them with calibrated images. Then the defocus measure can be used to estimate the depth of a scene. The coded aperture method [7] changes the shape of camera aperture to make defocus deblurring more reliable. A defocus map and an all-focused image can be obtained after deconvolution using calibrated blur kernels. These methods require additional illumination or camera modification to obtain a defocus map from a single image.

In this paper, we focus on a more challenging problem of recovering the defocus map from a single image captured by an uncalibrated conventional camera. Elder and Zucker [8] used the first and second order derivatives of the input image to find the locations and the blur amount of edges. The defocus map obtained is sparse. Bae et al. [9] extend this work and obtain a full defocus map from the sparse map using a interpolation method. Zhang and Cham [10] estimate the defocus map by fitting a well-paramerterized model to edges and use the defocus map to perform single image refocusing. The inverse diffusion method [11] models the defocus blur as a heat diffusion process and uses the inhomogeneous inversion heat diffusion to estimate defocus blur at edge locations. Tai and Brown [12] use local contrast prior to measure the defocus at each pixel and then apply MRF propagation to refine the defocus map. In contrast, we estimate the defocus map in a different but effective way. The input image is re-blurred using a known Gaussian blur kernel and the ratio between the gradients of input and re-blurred images is calculated. We show that the blur amount at edge locations can be derived from the ratio. We then formulate the blur propagation as a optimization problem. By solving the optimization problem, we finally obtain a full defocus map.

We propose an efficient blur estimation method based on the Gaussian gradient ratio, and show that it is robust to noise, inaccurate edge location and interference from neighboring edges. Without any modification to cameras or using additional illumination, our method is able to obtain the defocus map of a single image captured by conventional camera. As shown in Fig. 1, our method can estimate the defocus map of the scene with fairly good extent of accuracy.



(a)

(b)

Figure 2: Thin lens model. (a) Focus and defocus for thin lens model. (b) The diameter of CoC c as a function of the object distance d and f-stop number N given $d_f = 500mm$, $f_0 = 80mm$.

2. Defocus Model

We estimate the defocus blur at edge locations. As step edge is the main edge type in natural images, we consider only step edges in this paper. An ideal step edge can be modeled as:

$$f(x) = Au(x) + B, \tag{1}$$

where $u(x)$ is the step function. A and B are the amplitude and offset of the edge respectively. Note that the edge is located at $x = 0$.

We assume that focus and defocus obey the thin lens model [13]. When an object is placed at the focus distance d_f, all the rays from a point of the object will converge to a single sensor point and the image will appear sharp. Rays from a point of another object at distance d will reach multiple sensor points and result in a blurred image. The blurred pattern depends on the shape of aperture and is called the circle of confusion (CoC) [13]. The

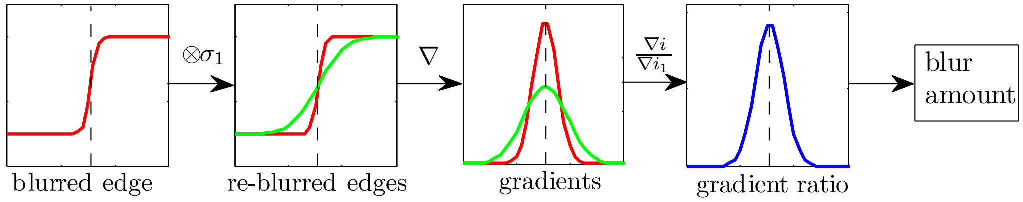


Figure 3: The overview of our blur estimation approach: here, $\otimes$ and ∇ are the convolution and gradient operators respectively. The black dash line denotes the edge location.

diameter of CoC characterizes the amount of defocus and can be written as

$$c = \frac{|d - d_f|}{d} \frac{f_0^2}{N\left(d_f - f_0\right)}, \tag{2}$$

where f_0 and N are the focal length and the stop number of the camera respectively. Fig. 2 illustrate focus and defocus for thin lens model and how the diameter of circle of confusion changes with d and N, given fixed f_0 and d_f. As we can see, the diameter of the CoC c is a non-linear monotonically increasing function of the object distance d.

The defocus blur can be modeled as a convolution of a sharp image with the point spread function (PSF). The PSF is usually approximated by a Gaussian function $g(x, \sigma)$, where the standard deviation $\sigma = kc$ measures the defocus blur amount and is proportional to the diameter of the CoC c. A blurred edge $i(x)$ is then given by,

$$i(x) = f(x) \otimes g(x, \sigma). \tag{3}$$

3. Defocus Blur Estimation

Fig. 3 shows the overview of our blur estimation method. An edge is re-blurred using a known Gaussian kernel. Then the ratio between the gradient magnitude of the step edge and its re-blurred version is calculated. The ratio is maximum at the edge location. Using the maximum value, we can compute the amount of the defocus blur at the edge location.

For convenience, we describe our blur estimation method for 1D case first and then extend it to 2D image. The gradient of the re-blurred edge is:

$$\begin{aligned} \nabla i_1(x) &= \nabla\left(i(x) \otimes g(x,\sigma_0)\right) \\ &= \nabla\left((Au(x)+B) \otimes g(x,\sigma) \otimes g(x,\sigma_0)\right) \\ &= \frac{A}{\sqrt{2\pi(\sigma^2+\sigma_0^2)}} \exp\left(-\frac{x^2}{2(\sigma^2+\sigma_0^2)}\right), \end{aligned} \tag{4}$$

where σ_0 is the standard deviation of the re-blur Gaussian kernel. We call it the re-blur scale. The gradient magnitude ratio between the original and re-blurred edges is

$$\frac{|\nabla i(x)|}{|\nabla i_1(x)|} = \sqrt{\frac{\sigma^2+\sigma_0^2}{\sigma^2}} \exp\left(-\left(\frac{x^2}{2\sigma^2} - \frac{x^2}{2(\sigma^2+\sigma_0^2)}\right)\right). \tag{5}$$

It can be proved that the ratio is maximum at the edge location ($x = 0$) and the maximum value is given by:

$$R = \frac{|\nabla i(0)|}{|\nabla i_1(0)|} = \sqrt{\frac{\sigma^2+\sigma_0^2}{\sigma^2}}. \tag{6}$$

Giving an insight on (4) and (6), we notice that the edge gradient depends on both the edge amplitude A and blur amount σ, while the maximum of the gradient magnitude ratio R eliminates the effect of edge amplitude A and depends only on σ and σ_0. Thus, given the maximum value R at the edge locations, the unknown blur amount σ can be calculated using

$$\sigma = \frac{1}{\sqrt{R^2-1}}\sigma_0. \tag{7}$$

For 2D images, the blur estimation is similar. We use 2D isotropic Gaussian kernel for re-blurring and the gradient magnitude can be computed as follows:

$$\|\nabla i(x,y)\| = \sqrt{\nabla i_x^2 + \nabla i_y^2} \tag{8}$$

where ∇i_x and ∇i_y are the gradients along x and y directions respectively. In our implementation, We set the re-blurring $\sigma_0 = 1$ and use Canny edge detector [14] to perform the edge detection. In this work, we also assume the camera response curve is linear.

The blur scales are estimated at each edge location, forming a sparse depth map denoted by $\hat{d}(x)$. However, quantization error at weak edges, noise or soft shadows may cause inaccurate blur estimates at some edge locations. To solve this problem, we apply joint bilateral filtering (JBF) [15] on the sparse depth map $\hat{d}(x)$ to refine those inaccurate blur estimates. By using the original input image I as the reference, the filtered sparse defocus map can be defined as:

$$BF(\hat{d}(x)) = \frac{1}{W(x)} \sum_{y \in \mathcal{N}(x)} G_{\sigma s}(\|x-y\|) G_{\sigma_r}(\|I(x) - I(j)\|) \hat{d}(y) \tag{9}$$

where $W(x)$ is the normalization factor and $\mathcal{N}(x)$ is the neighborhood of x given by the size of spatial Gaussian filter G_{σ_s}. σ_s controls the size of the spatial neighborhood and σ_r controls the influence of intensity difference. We set them to be 10% of the image size and 10% of the intensity range, respectively. Note that the filtering is only performed on the edge locations. As we can see from Fig. 4, the joint bilateral filtering corrects some errors in the sparse defocus map, and thus avoids the propagation of the errors in defocus map interpolation described in the next section.

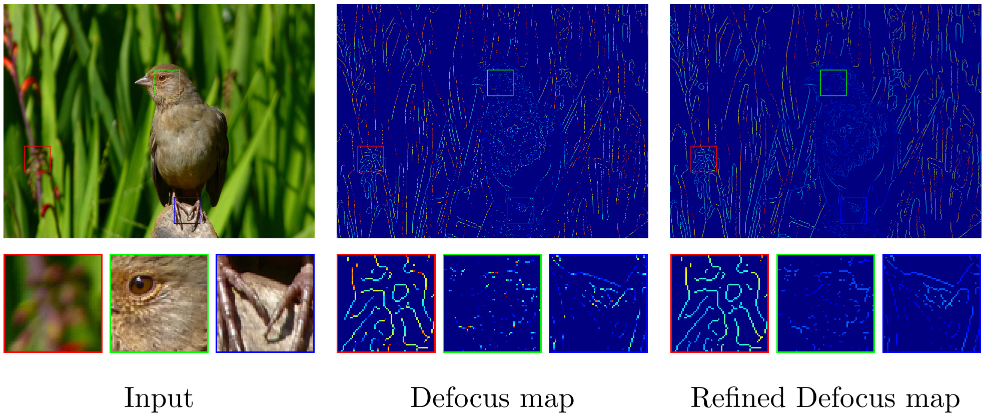


Figure 4: Defocus map refinement using joint bilateral filtering. The joint bilateral filtering correct defocus estimation errors caused by noise or soft shadows.

4. Defocus Map Interpolation

Our defocus blur estimation method describe in previous step produces a sparse defocus map $\hat{d}(x)$. In this section, we provided a way to propagate the defocus blur estimates from edge locations to the entire image and obtain a full depth map $d(x)$. To achieve this, we want to seek a defocus map $d(x)$ which is close to the sparse defocus map $\hat{d}(x)$ at each edge location. Furthermore, we prefer the defocus blur discontinuities to be aligned with image edges. Edge-aware interpolation methods [16, 17] are usually used for these tasks. Here, we apply the matting Laplacian [18] to perform the defocus map interpolation. Formally, the depth interpolation problem can be formulated as minimizing the following cost function:

$$E(d) = d^T L d + \lambda (d - \hat{d})^T D (d - \hat{d}), \tag{10}$$

where $\hat{d}$ and d are the vector forms of the sparse defocus map $\hat{d}(x)$ and the full defocus map $d(x)$ respectively. L is the matting Laplacian matrix and D

is a diagonal matrix whose element D_{ii} is 1 if pixel i is at the edge location, and 0 otherwise. The scalar λ balance between fidelity to the sparse depth map and smoothness of interpolation. The (i, j) element of L is defined as:

$$\sum_{k|(i,j)\in\omega_k} \left(\delta_{ij} - \frac{1}{|\omega_k|}\left(1 + (I_i - \mu_k)^T(\Sigma_k - \frac{\epsilon}{|\omega_k|U_3}^{-1}(I_j - \mu_K)\right)\right), \tag{11}$$

where δ_{ij} is the Kronecker delta, U_3 is a 3×3 identity matrix, μ_k and σ_k are the mean and covariance matrix of the colors in window ω_k. I_i and I_j are the colors of the input image I at pixel i and j respectively. ϵ is a regularization parameter and $|\omega_k|$ is the size the window ω_k. For the detailed derivation of Eq. 11, readers can refer to [18].

The optimal d can be obtained by solving the following sparse linear system:

$$(L + \lambda D)d = \lambda D\hat{d}. \tag{12}$$

In our implementation, we use a fixed λ values 0.005, so that a soft constraints is put on d to further refine small errors in our blur estimation. Such a soft matting method are also applied in [19, 20] to deal with dehazing and spatially variant white balance problems.

5. Experiments

We first test the robustness of our method on synthetic images. We synthesize a set of bar images, one of which is shown in Fig. 5(a). The blur amount of the edge increases linearly from 0 to 5. Under noise conditions, as show in Fig.5(d), our method can achieve a reliable estimation. And we also find that our blur estimation result of edges with smaller blur amounts are less affected by noise compared with those with larger blur amounts.

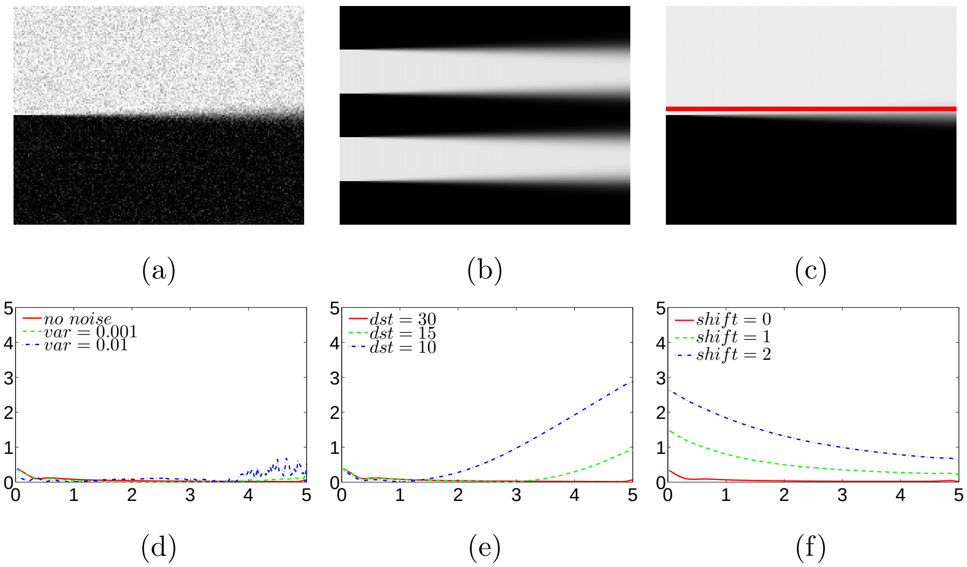


Figure 5: Performance of our blur estimation method on synthetic images. (a) The synthetic image with noise ($var = 0.01$). (b) The synthesis image with edge distance of 20 pixels. (c) The synthesis image and edge detection with edge shifted by 2 pixels. (d) Estimation errors under noise condition. (e) Estimation errors with difference edge distances. (f) Estimation errors with edge mis-localization. The x and y axes are the blur amount and corresponding estimation error respectively.

We also test our blur estimation on bar images with different edge distances. Fig. 5(e) shows that our blur estimation result is affected by neighboring edges, especially when the edge distance is small and the blur amount is large, but the estimation errors are controlled in a low level if the blur amount is not large (< 3).

In Fig. 5(c), we shift the detected edges to simulate edge mis-localization. The result is shown in Fig. 5(f). Our edge estimation is robust to edge mis-localization for edge with large blur amount, while it may cause large errors for the blur estimation of sharp edges. However, in practice, the sharp edges

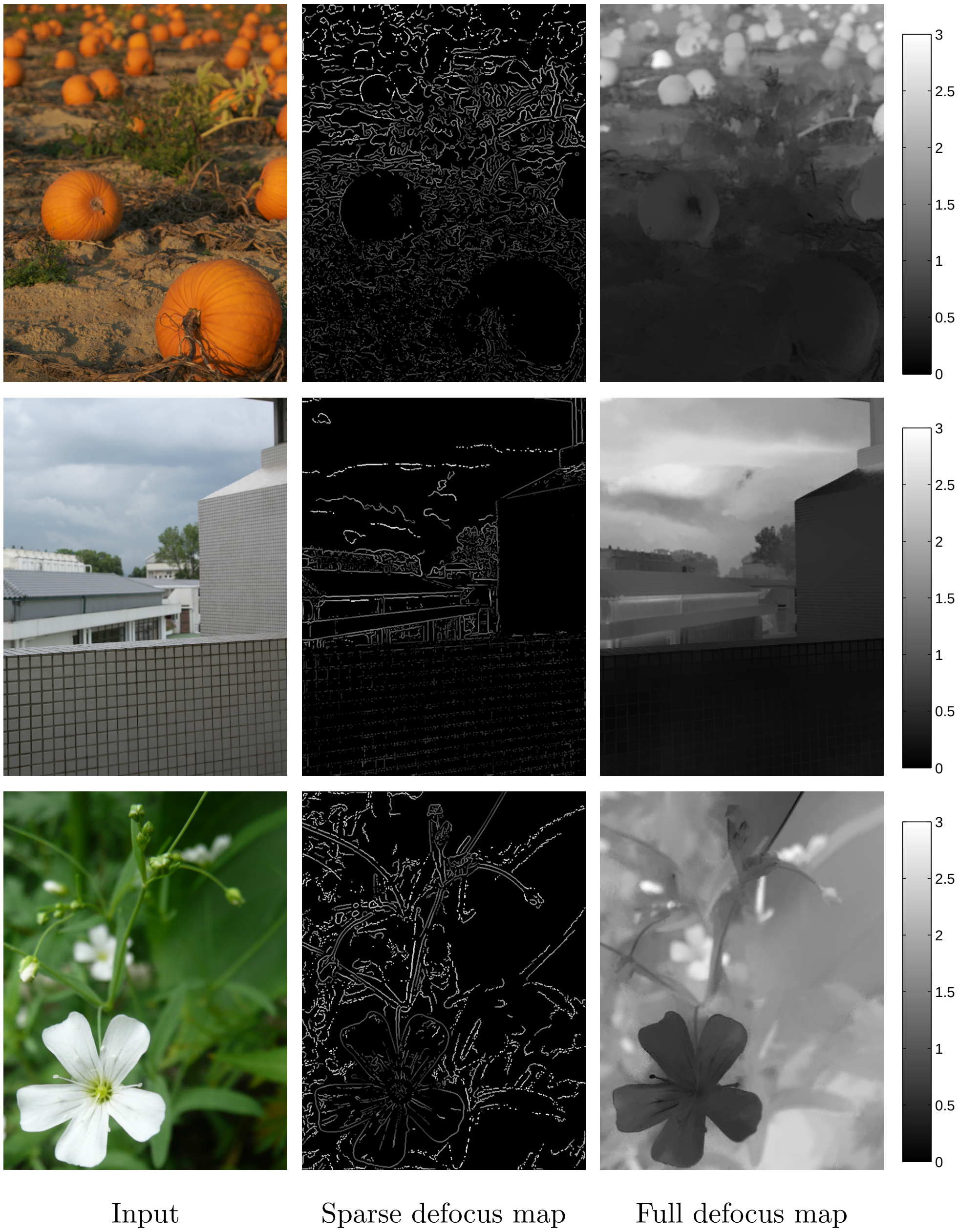
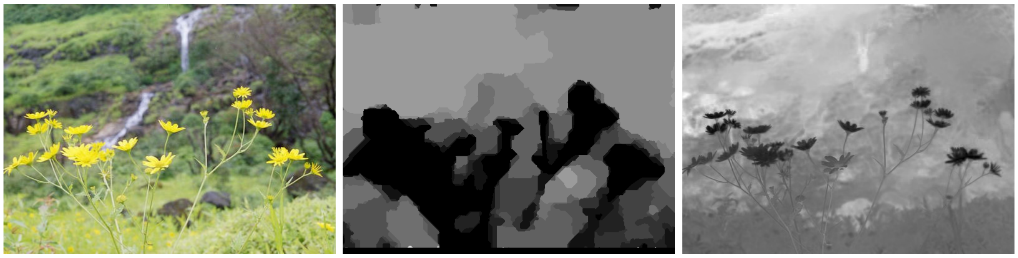


Figure 6: Defocus map estimation on real images. Our method can work on different types of scenes with continuous depth (the pumpkin image) and layered depth (the building image and the flower image), resulting in defocus maps with fairly good extend of accuracy.



Input image Inverse diffusion method Our result

Figure 7: Comparison of our method with the inverse diffusion method. Our result is able to recover a visually more plausible and more accurate defocus map. For example, the flower layer in the image is better separated from the background layer in our result.

usually can be located very accurately by edge detection methods, which greatly reduces the estimation error.

As show in Fig. 6, we test our method on some real images. In the pumpkin image, the depth of the scene changes continuously from the bottom to the top of the image. The estimated defocus map captures the continuous change of the depth. In the building image, the scene mainly contains three layers: the wall, house and sky layers. Our method is able to produce defocus maps corresponding to those layers. The flow image gives the similar result. One more example is shown in Fig. 1. The defocus map capture the foreground boy layer and the continuous change of the background. As we can see from these results, our method is able to recover a reasonably good defocus map from a single image.

In Fig. 7, we compare our method with the inverse diffusion method [11]. The inverse diffusion method produces coarse defocus map. the flower layer is not well separated with background layers and contains some error estimates. In contrast, our method is able to produce a more accurate and continuous

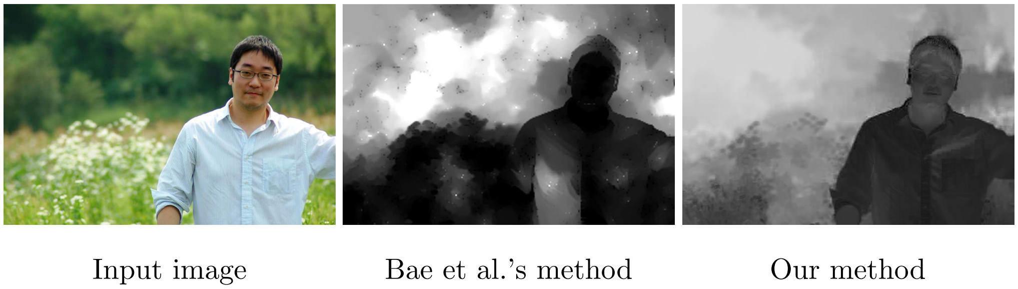
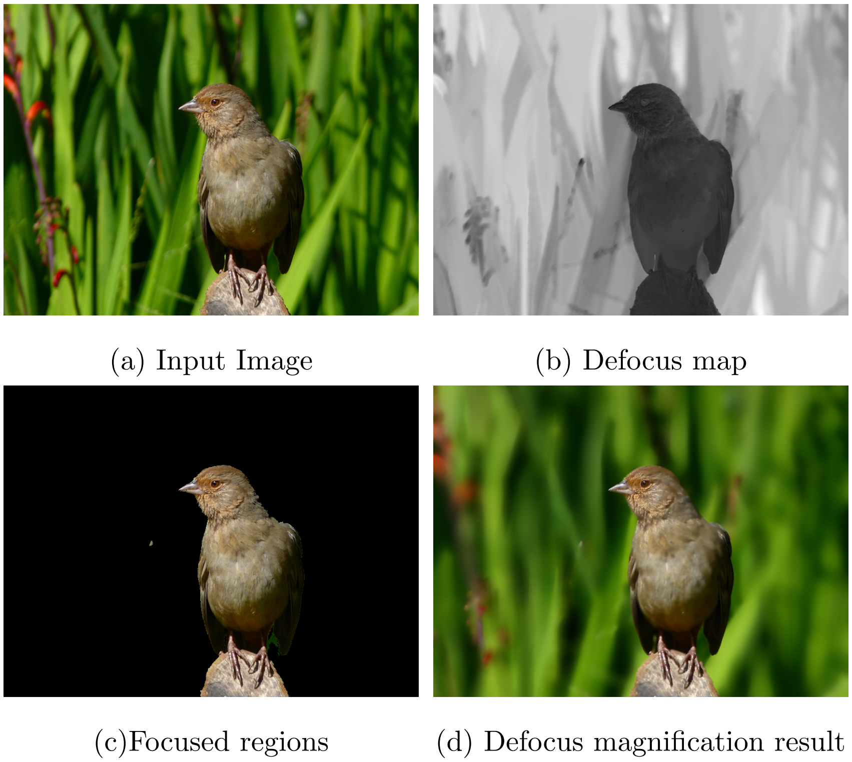


Figure 8: Comparison of our method with Bae et al.'s method. Our method contains less noise and better captures the depth change of the scene.

defocus map. In our defocus map, the flower is well separated with the background.

A comparison of our method with Bae et al.'s method [9] is shown in Fig. 8. While both method use joint bilateral filtering to refine sparse defocus maps, the result of Bae et al.'s method still contains some visible defocus estimation errors (the white noisy points) in the final full defocus map. However, our result is more accurate and the depth change of the scene are well captured by the defocus map.

Our method can be used to extract focused regions from defocued images. After defocus map interpolation, a pixel is assigned to be focused if its defocus value is smaller than a threshold t_f(typically, $t_f = 1$). As shown in Fig. 9(c), our approach is successful in segmenting the focused regions from the image. Our method is especially useful if the foreground and background contains similar colors, in which the segmentation or matting methods may fail to extract the region of interest. As illustrated in Eq. 2, the size of defocus blur is proportional to aperture size. Thus, we can linearly increase the defocus map values to simulate a larger aperture effect. Fig. 9(d) shows the defocus


(a) Input Image

(b) Defocus map

(c)Focused regions

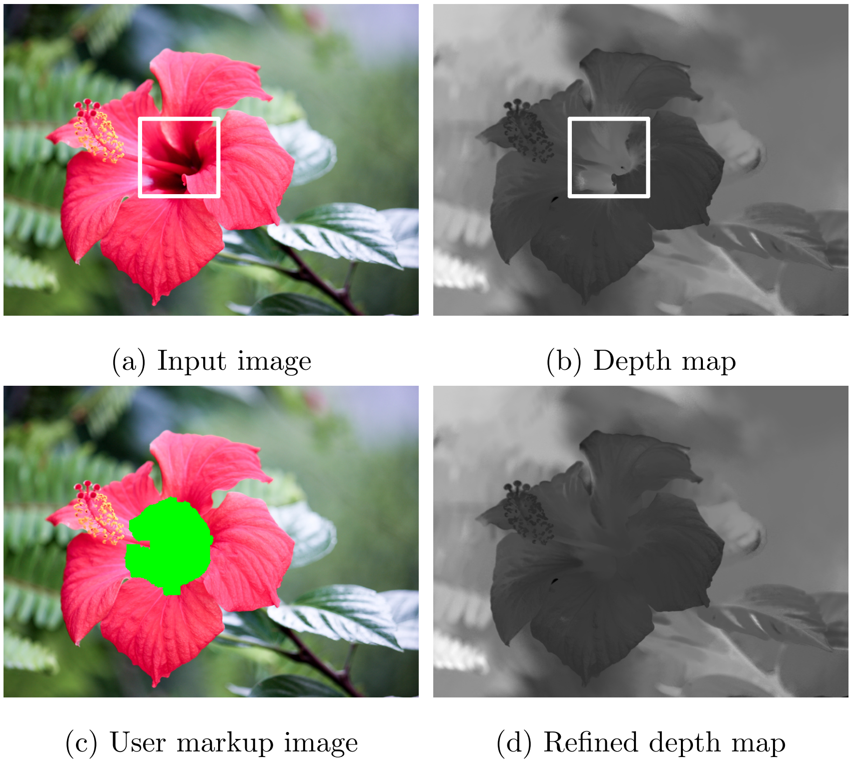
(d) Defocus magnification result

Figure 9: Applications of our method. Our method can be use to extract focused regions in the image and perform defocus magnification to emphasis the main subject in the image.

magnification result. We can see that the bird in the image remains sharp while the background is blurred more. The defocus magnification is able to avoid distracting background and emphasis the main subjects in the image.

6. Limitations and Discussions

Blur Texture Ambiguity. One limitation of our blur estimation is that it can not tell whether a blur edge is caused by defocus or blur texture (soft shadows or blur patterns) of the input image. For the later case, the defocus value we obtained is a measurement of the sharpness of the edge.



(a) Input image

(b) Depth map

(c) User markup image

(d) Refined depth map

Figure 10: The blur texture ambiguity. This ambiguity can cause some errors in our defocus map. The error region is shown in the white rectangle. These errors can be corrected by providing a user markup image to exclude the blur estimates in that region.

It is not the actual defocus value of the edge. This ambiguity may cause some artifacts in our result. One example is shown in Fig. 10. The region indicated by the white rectangle is actually blur texture of the flower, but our method treats it as defocus blur, which results in error defocus estimation in that region. Additional images [1, 5] are usually used to remove the blur texture ambiguity. Here, we introduce the user interaction to handle this problem. A user can mark the blur texture region to exclude the estimated defocus values in those regions, so that the defocus values are propagated from reliable neighboring regions. As shown in Fig. 10(d), the blur texture ambiguity can be properly handled by using the user interaction.

Defocus map and Depth. If the camera settings are provided, the defocus map can be converted to depth map. However, there is a focal plane ambiguity in defocus blur and depth mapping. When an object appears blur in the image, it can be on either side of the focal plane. To remove this ambiguity, most depth from defocus methods assume all objects of interest are located on one side of the focal plane and put the focus point on the nearest/farthest point in the scene.

7. Conclusion

In this paper, we show that the defocus map can be recovered from a single image. A new method is presented to estimate the blur amount at edge locations based on the Gaussian gradient ratio. A full defocus map is then produced using the matting interpolation. We show that our method is robust to noise, inaccurate edge location and interferences of neighboring edges and is able to generate more accurate defocus maps compared with existing methods. We also discuss the blur texture ambiguity arising in recovering defocus map from a single image and the focal plane ambiguity when converting defocus map to depth map. We also propose some possible ways to remove those ambiguities. In the future, we would like to extend our method to work on more edge types and apply it on other problems such as motion blur estimation.

Acknowledgment

We thank the reviewers for helping to improve this paper. We thank Xiaopeng Zhang, Dong Guo and Ning Ye for their discussion and useful suggestion. This work is supported by NUS Research Grant #R-252-000-383-112.

References

[1] P. Favaro, S. Soatto, A geometric approach to shape from defocus, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 406–417.

[2] P. Favaro, S. Soatto, M. Burger, S. Osher, Shape from defocus via diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 30 (3) (2008) 518–531.

[3] A. P. Pentland, A new sense for depth of field, IEEE Trans. Pattern Anal. Mach. Intell. 9 (4) (1987) 523–531.

[4] C. Zhou, O. Cossairt, S. Nayar, Depth from Diffusion, in: Proc. CVPR, 2010, pp. 1110 – 1117.

[5] C. Zhou, S. Lin, S. K. Nayar, Coded Aperture Pairs for Depth from Defocus, in: Proc. ICCV, 2009, pp. 325 – 332.

[6] F. Moreno-Noguer, P. N. Belhumeur, S. K. Nayar, Active refocusing of images and videos, ACM Trans. on Graphics 26 (3) (2007) 67–75.

[7] A. Levin, R. Fergus, F. Durand, W. T. Freeman, Image and depth from a conventional camera with a coded aperture, ACM Trans. on Graphics 26 (3) (2007) 70–78.

[8] J. Elder, S. Zucker, Local scale control for edge detection and blur estimation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (7) (1998) 699–716.

[9] S. Bae, F. Durand, Defocus magnification, Proc. Eurographics (2007) 571–579.

[10] W. Zhang, W.-K. Cham, Single image focus editing, in: ICCV WorkShop, 2009, pp. 1947 –1954.

[11] V. P. Namboodiri, S. Chaudhuri, Recovery of relative depth from a single observation using an uncalibrated (real-aperture) camera, in: Proc. CVPR, 2008, pp. 1–6.

[12] Y.-W. Tai, M. S. Brown, Single image defocus map estimation using local contrast prior, in: Proc. ICIP, 2009.

[13] E. Hecht, Optics (4th Edition), Addison Wesley, 2001.

[14] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.

[15] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, K. Toyama, Digital photography with flash and no-flash image pairs, ACM Trans. on Graphics 23 (3) (2004) 664–672.

[16] A. Levin, D. Lischinski, Y. Weiss, Colorization using optimization, ACM Trans. on Graphics 23 (3) (2004) 689–694.

[17] D. Lischinski, Z. Farbman, M. Uyttendaele, R. Szeliski, Interactive local adjustment of tonal values, in: ACM Trans. on Graphics, 2006, pp. 646–653.

[18] A. Levin, D. Lischinski, Y. Weiss, A closed-form solution to natural image matting, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2) (2008) 228–242.

[19] K. He, J. Sun, X. Tang, Single image haze removal using dark channel prior, IEEE Trans. Pattern Anal. Mach. Intell. 99.

[20] E. Hsu, T. Mertens, S. Paris, S. Avidan, F. Durand, Light mixture estimation for spatially varying white balance, ACM Trans. on Graphics 27 (2008) 70:1–70:7.