Bokeh Effect

Harshal Mittal
Dept. of Electronics Engg.
IIT Roorkee
16116020
hmittal@ec.iitr.ac.in

Harshit Bansal
Dept. of Electronics Engg.
IIT Roorkee
16116021
hbansal@ec.iitr.ac.in

Shubham Maheshwari
Dept. of Electronics Engg.
IIT Roorkee
16116065
smaheshwari@ec.iitr.ac.in

***Abstract*— In this project, we aim to produce a Bokeh effect from a single image. The main principle behind this effect is depth mapping. Depth mapping is a prediction of depth at each point in the image. Thus, given a depth mapping, and a pre-defined threshold, we can approximately classify the points into foreground and background. The background points would be applied a Gaussian kernel, producing a blur effect, while the foreground mask can be simply applied to the original image. The two would then be concatenated to produce the final Bokeh image. Also, the depth map of an image is directly related to its defocus map, given the camera parameters. However, due to the focal plane ambiguity, it is generally assumed that all de-focussed pixels lie on one side of the focal plane and the focussed on the other side, producing a rough depth map. Thus the problem reduces to that of defocus map estimation, which has been extensively discussed in the past.**

I. INTRODUCTION

Defocus blur estimation plays an important role in many computer vision and computer graphics applications including depth estimation, image deblurring and refocusing. Previously, the problem has been tackled by taking two images at varying depths or through the use of portrait lenses. However, recent attention has been shifted towards depth estimation from a single image taken from an ordinary camera. The project aims to utilize defocus blur estimation to produce a Bokeh effect from a single image. This is done by segmenting the image into two parts: foreground and background based on the value of the defocus blur, and further applying a Gaussian kernel to the background. Defocus blur is estimated from the ratio of the magnitude of the gradients of the original and the re-blurred image.

The overall flow of the report can be described as follows: In section 2, the defocus model is introduced. In section 3, the overall procedure of the algorithm is explained followed by the results in section 4. Finally, the project report ends with some limitations of the program and the conclusion.

II. DEFOCUS MODEL

When an image is taken from a camera, only the points lying on the focal plane of the camera would be on focus. The rest of the points would have more than one mapping on the screen producing a blur effect.

Assuming that the focus and defocus obey the thin lens model, when an object is placed at distance d_{f} from the lens, all the rays emerging from it would be converged at a single point on the sensor by the lens, thus producing a sharp image. While if the object is at a distance d from the lens, the rays produced by it will reach multiple sensor points on passing through the lens, and would result in a blurred image. This is shown in Fig 1. The blur pattern on the sensor plane depends on the lens aperture, and is called the Circle of Confusion (CoC), whose diameter can be approximated as :

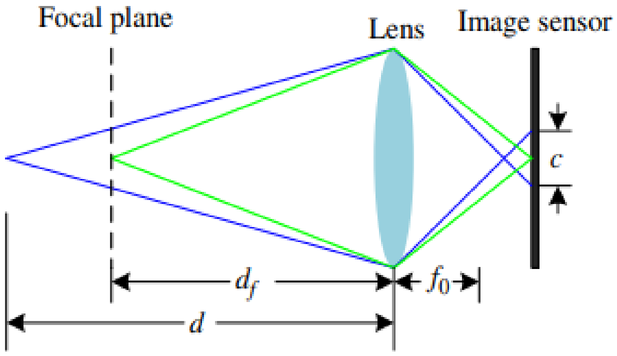

Fig. 1: Focus and defocus for thin lens model.

$$c = \frac{|d - d_{\mathrm{f}}|}{d} \frac{{f_{\mathrm{o}}}^2}{N(d - d_{\mathrm{f}})} \tag{1}$$

where σ = kc measures the defocus blur amount at any point and is proportional to the diameter of CoC. It is further explained in the next section.

III. PROCEDURE

A. Extracting the sparse defocus map

The defocus blur is taken only at the edges present in the image, which can be represented ideally as:

$$f(x) = Au(x) + B \tag{2}$$

The defocus blur can be understood as applying the convolution operator on the edge and the PSF (Point Spread Function), where the PSF is approximated by a Gaussian Function g(x,σ) The σ here is the measure of the defocus blur amount, and σ = kc (relating it to the CoC as mentioned previously). The blurred edge can then be written as :

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

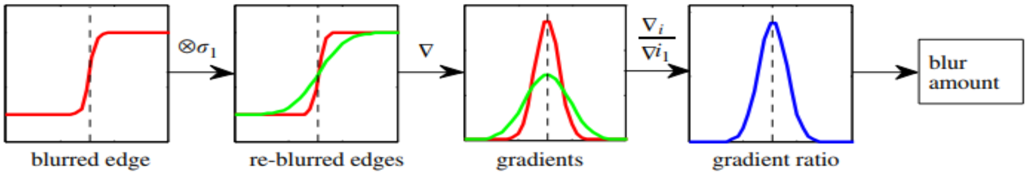

Fig. 2: The overview of our blur estimation approach: here, $*$ and ∇ are the convolution and gradient operators, respectively. The black dash line denotes the edge location at x=0.

We estimate the defocus blur (estimated only at edges) amount by first calculating the ratio between the gradient magnitudes in the original image and the re-blurred image at the edges as shown in Fig 2. We call this ratio as R and it would be higher at the edges compared to other points in the image, as expected. This is because the effect of blurring would be more pronounced at the edges compared to other image points, as the other points were already a bit soft and not so sharp, while the edges were smudge-free in the original image. This can also be proved mathematically as follows :

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

Here, σ_0 is the standard deviation of the re-blur Gaussian kernel.

The ratio of the gradient magnitudes between the original and re-blurred image is calculated as:

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

The ratio is maximum at the edge location(x = 0) and the maximun value is given by :

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

We can see that the maximum of the gradient magnitude ratio R eliminates the effect of edge amplitude A, and depends only on original image blur σ and the re-blur amount σ_0.

This was the case for 1D, blur estimation for 2D images is done along the same lines. A 2D isotropic Gaussian kernel is used for re-blurring and the gradient magnitude can be calculated as:

$$||\nabla i(x,y)|| = \sqrt{\nabla i_x^2 + \nabla i_y^2} \tag{7}$$

Once R is obtained, the unknown blur amount σ of the original image can be obtained as:

$$\sigma = \frac{1}{\sqrt{R^2-1}}\sigma_0 \tag{8}$$

In our implementation, we found out the re-blur amount σ_0 to be 1 by trying out a few different values, as $\sigma_0 = 1$ was giving acceptable results. As mentioned, the defocus blur is estimated only at edges. So σ at other points in the image is made 0 by first using Canny-Edge detection to obtain the edge points, and then applying a mask (comprising of non-edge points) to the defocus blur map to set the values to 0. This sparse defocus map (or sparse depth map) is denoted by $\hat{\text{d}}$(x). To remove the inaccuracies present in it due to weak or noisy edges, a Joint Bilateral Filter (JBF) is applied to refine the sparse defocus map.

B. *Bilateral Filtering*

The defocus blur can be understood as applying the convolution operator on the edge and the PSF (Point Spread Function), where the PSF is approximated by a Gaussian Function The sparse defocus blur map, which has been obtained at edge locations may contain some inaccuracies due to the quantization error at weak edges and noise or soft shadows in the image. Thus, joint bilateral filtering is applied to the sparse depth map to correct the wrong blur estimates. It thereby avoids the propagation of the errors in the defocus map interpolation as explained in the next section.

C. *Defocus Map Interpolation*

The defocus blur estimation method produces a sparse depth map which needs to be propagated to the full image to obtain the full depth map. Matting Laplacian, an edge-aware interpolation method is used to do the defocus map interpolation. It seeks to produce a depth map close to the sparse depth map and also aligning the discontinuities with the image edges. Overall, it aims to minimize the following loss function:

$$E(d) = d^{\mathrm{T}}Ld + \lambda(d-\hat{d})^{\mathrm{T}}D(d-\hat{d}) \tag{9}$$

where $\hat{d}$ and d are the vector forms of the sparse defocus map and the full defocus map 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 optimal d can be obtained by solving the following sparse linear system:

$$(L+\lambda D)d = \lambda D\hat{d} \tag{10}$$

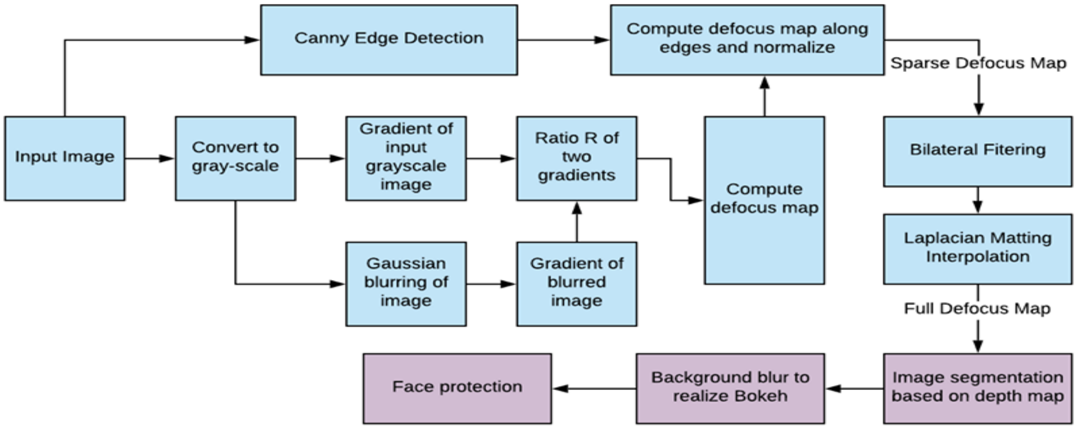
Canny Edge Detection
Compute defocus map along edges and normalize
Sparse Defocus Map
Input Image
Convert to gray-scale
Gradient of input grayscale image
Ratio R of two gradients
Compute defocus map
Bilateral Fitering
Gaussian blurring of image
Gradient of blurred image
Laplacian Matting Interpolation
Full Defocus Map
Face protection
Background blur to realize Bokeh
Image segmentation based on depth map

Fig. 3: Algorithm Flowchart.

D. *Bokeh Effect*

To achieve Bokeh effect after the full depth map estimation, the image can be segmented into two segmentations i.e., foreground and background. This would be done by deciding an appropriate threshold and considering all the points with their depth/defocus blur below the threshold as a part of the foreground and the remaining as background. Thus, the bit masks of these segmentations would be obtained. The bit mask of the foreground would be applied on the original image and the bit mask of the background would be applied on the blurred image which is obtained by applying a Gaussian kernel to the original image. Finally, both images are combined to produce the final image with the Bokeh effect. Due to the imperfect segmentations, some parts of the face like eyebrow, hair are usually blurred as they are segmented into the background. To resolve this problem, face detection can be used to avoid blurring of the face. If the face is detected on the original image using Matlab Computer Vision Toolbox and the average depth (or defocus blur) of the face region is small, no blur would be applied regardless of the result. The position and size of the rectangular box, returned by face detection algorithm of Matlab Computer Vision Toolbox, need to be adjusted as this box region only covers the face but excludes the hair.

The flowchart of the overall algorithm is presented in Figure 3.

IV. Result

The results are good when sharp edges are present for the foreground objects and some amount of defocus blur is present for the background objects. This is because the algorithm for the estimation of the sparse defocus map considers the ratio of the gradients of the original and the blurred image at the edges. A blurred background would have a relatively smaller gradient ratio, thus assigning it a high defocus blur. Due to the same reason, sharp edges would have a higher gradient ratio and thus a small defocus blur. The joint bilateral filtering algorithm also results in a more continuous and accurate sparse defocus map. The face detection algorithm further improves the results whenever the algorithm blurs the facial parts.

The results of the algorithm on some images are shown in Figures 4, 5, and 6. As can be seen in Figure 4, the image consists of flowers in the foreground having sharp edges and a relatively blurred background. The algorithm successfully segments the foreground from the background and generates appreciable results. In Figure 5, the face of the person had some amount of blur, which was rectified by the face protection part. Also, an instance of the failure of the algorithm is depicted in Figure 6. Here, the image has a bigger depth map with background having no defocus blur. Thus, the algorithm fails to blur the background and incorrectly blurs the clothes of the person.

V. Limitations and Future Work

Defocus blur is more pronounced in cameras which have a shallow depth of field and is less pronounced in a camera with a larger depth of field. Generally, cheaper cameras have a larger depth of field whereas expensive cameras have a shallow depth of field. Hence this method would generate a poor result whenever the background of the image has less defocus blur. Another limitation is the non-smooth transition between foreground and background on the output image. This happens due to the abrupt shift in the levels of blurring between the two different segmentations. This can be partially resolved by implementing a more gradual and continuous Gaussian kernel along the points where the two segmentations are concatenated and we also intend to do as future work.

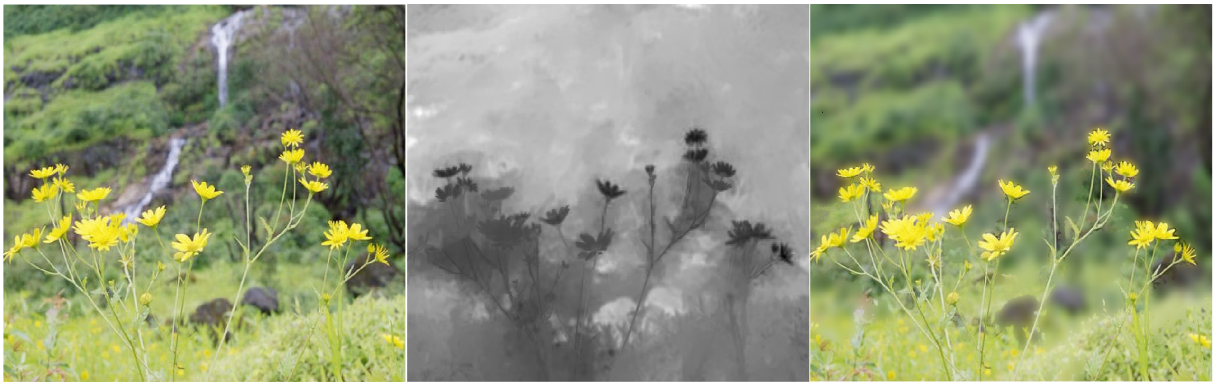

Fig. 4: Left: Original Image, Center: Extracted full Defocus map, Right: Resultant image after Bokeh Effect.

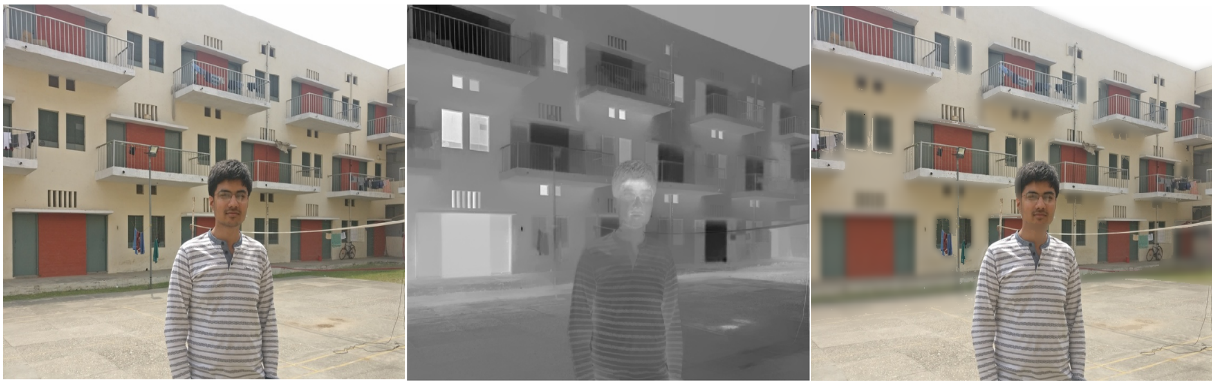

Fig. 5: Left: Original Image, Center: Extracted full Defocus map, Right: Resultant image after Bokeh Effect.

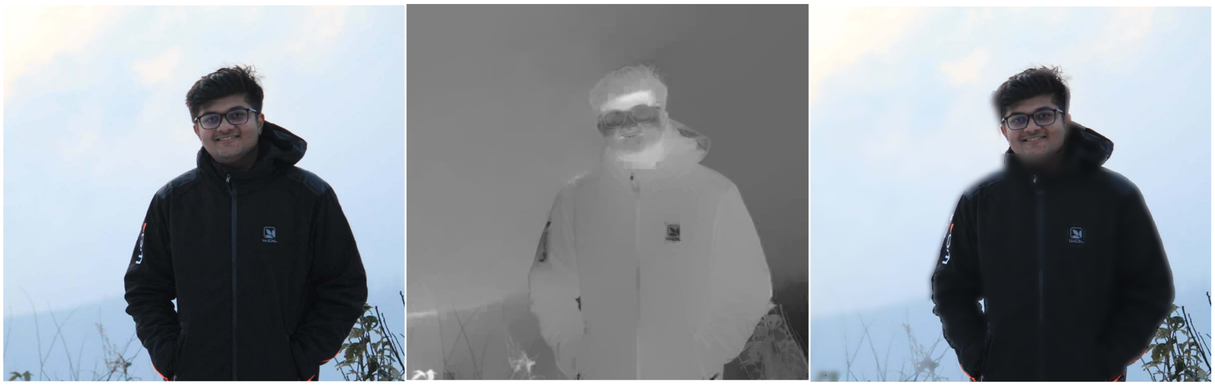

Fig. 6: Left: Original Image, Center: Extracted full Defocus map, Right: Resultant image after Bokeh Effect.

ACKNOWLEDGMENT

We would like to thank Professor Saumik Bhattacharya for providing us the opportunity to work on this project and guiding us throughout. We learnt a lot during the entire period, which was a very indulging experience and would also prove to be helpful further.

REFERENCES

[1] X. Zhu, S Cohen, S Schiller, and P. Millanfar Estimating Spatially Varying Defocus Blur. IEEE Transactions on, volume 22, pages 4879 4891, 2013.

[2] A. Chakrabarti, T. Zickler, and W. T. Freeman Analyzing spatially varying blur" in Proc. IEEE CVPR, Jun. 2010,pp. 25122519.

[3] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. ACM Trans. Graphics, 2007.

[4] P. Favaro, S. Soatto, A geometric approach to shape from defocus, IEEE Trans.Pattern Anal. Mach. Intell. 27 (3) (2005) 406417.

[5] A.P. Pentland, A new sense for depth of field, IEEE Trans. Pattern Anal. Mach. Intell. 9 (4) (1987) 523531.