Bokeh Rendering from Defocus Estimation

Xianrui Luo, Juewen Peng, Ke Xian(✉), Zijin Wu, and Zhiguo Cao

School of Artificial Intelligence and Automation,
Huazhong University of Science and Technology, Wuhan 430074, China
{xianruiluo,juewenpeng,kexian,zjwuzijin,zgcao}@hust.edu.cn

Abstract. In this paper, we study realistic bokeh rendering from a single all-in-focus image. Existing computational bokeh rendering methods generate bokeh effects by adding a simple flat background blur. As a result, the rendering results are different from the real bokeh on DSLR cameras. To address this issue, we propose a multi-stage network to learn shallow depth-of-field from a single bokeh-free image. In particular, our network consists of four modules: defocus estimation, radiance, rendering, and upsampling. The four modules are trained on different sizes to learn global features as well as local details around the boundaries of in-focus objects. Experimental results show that our approach is capable of rendering a pleasing distinctive bokeh effect in complex scenes.

Keywords: Bokeh rendering · Defocus estimation · Radiance · Upsampling

1 Introduction

Realistic bokeh effect, *a.k.a.* shallow depth-of-field, is an important feature in photography. It is often rendered from a digital single-lens reflex camera (DSLR) with a wide aperture by maneuver operations, however, the cost of time and money makes this process unfriendly to non-professionals. It is expected that we are able to produce bokeh effect from a single image with narrow aperture without the expensive hardware of DSLR cameras.

In this work, we propose a multi-stage network to learn bokeh synthesis by means of defocus estimation only supervised by bokeh images (Fig. 1). We employ an effective monocular depth estimation model as the basis of defocus estimation model, however we make modifications at the end of the fully convolutional network at the first stage because it has difficulty in learning without the ground truth of defocus maps. The modified output is changed from single-channel to multi-channel and it represents a group of probabilistic defocus maps for different blur amount. The final result is a combination of layered bokeh

X. Luo, J. Peng—Equal contributions.

A. Bartoli and A. Fusiello (Eds.): ECCV 2020 Workshops, LNCS 12537, pp. 245–261, 2020.
https://doi.org/10.1007/978-3-030-67070-2_15

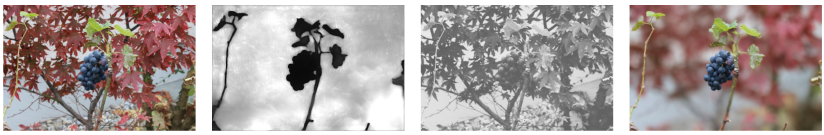

Fig. 1. Given a single narrow aperture image, our algorithm estimates the corresponding defocus map using a network based on depth estimation, and a weight map from radiance module which demonstrates the brightness relationship between pixels is predicted. This network was trained only under the supervision of bokeh images with a wider aperture size.

blurred by kernels with different radius. We change the output back to a single-channel defocus map at the third stage.

Furthermore, we apply a simple radiance module to learn the relationship between the intensity of pixels in a single image, because we aim to produce the Circle of Confusion (CoC), which is crucial for the aesthetic quality of shallow depth-of-field effects. Distinctive CoC only appears when the intensity of pixel is much higher than any other pixels, which means the real value of the pixels are only available in High Dynamic Range (HDR). Since we process bokeh-free images under Low Dynamic Range, we predict weight maps to highlight the pixels with high intensity.

To train our network in a multi-stage manner, low resolution images of 1/4 the original size are used for input for the first stage to obtain global information, and at the second stage the model gets the upsampled features from the first stage and use high resolution images of 1/2 resolution to refine the details around edges. For stage 3, in order to present the CoC effect distinctively, the output of last layer is changed to a single-channel defocus map, which is further divided into layers and rendered on the guidance of Proportional relationship between blur radius and value within the map. For the last upsampling stage, first bilinear interpolation is used on rendered bokeh, then we adopt mask from predicted single-channel defocus map to make the in-focus objects clearer.

In summary, our main contributions are:

(1) We propose an autofocus model for bokeh rendering from monocular images with narrow aperture.
(2) We utilize defocus estimation and radiance estimation to predict blur amount and the intensity relationship among pixels in each image.
(3) A multi-stage training scheme using a combination of low resolution and high resolution images to acquire global features as well as preserving the details around boundaries.

2 Related Work

2.1 Defocus Estimation

Defocus maps can be applied in various tasks such as image deblurring [45], blur magnification [1] and depth estimation [24,28]. Defocus maps estimation can be

categorized into two types of methods. The first category is region-based, and the other one is edge-based.

Region based methods use image patches to estimate the defocus amount in a straightforward manner. There are works [28,29] that focus on detecting and estimating defocus map from images that have small blur amount. Yan *et al.* [43] apply a regression neural network to predict the blur type and its parameters. Tang *et al.* [34] use log averaged spectrum residual to obtain a coarse defocus map and refine it iteratively by exploiting the relevance of similar neighbor regions.

As for Edge-based methods, Zhuo *et al.* [47] introduce a typical method to use the ratio of the gradients of original and blurred images at edge points to produce a sparse map, and Laplacian matting [20] is used for interpolation with the input image as guidance. Park *et al.* [25] introduce various hand-crafted features and a deep convolutional neural network is applied to learn deep features and produce defocus map from multi-scale image patches under the guidance of edge-preserved images. However, sometimes the value in homogeneous areas is inconsistent. Xu *et al.* [42] propose a metric based on the connection between the metric rank of a defocused patch and the amount of blur at edges, then a sparse defocus map is reconstructed from ranks of local patches in gradient domain. There are also approaches to exploit the frequency information of image edges for defocus estimation, using spectrum contrast [33] or sub-band decomposition[3].

2.2 Monocular Depth Estimation

Monocular depth estimation is an important area which has potential in various tasks. Eigen *et al.* [7] propose a multi-scale deep neural network based on AlexNet [18] and extend the work [6] by replacing AlexNet with a deeper VGG16 [31] to increase accuracy. At present, the commonly used method to obtain the depth is from kinect infrared sensor (NYU Depth V2 [30]) or Lidar (KITTI [8]). But in reality, it is not always easy to obtain the depth value corresponding to the scene. So unsupervised methods are purposed, using stereo matching [10] or left-right consistency loss [9]. Apart from depth maps as ground truth, shallow depth-of-field images can also be used as supervision [32]. Recently, internet photos [22,39,40], which are promising sources of supervision, have attracted more and more attention. The models trained on this kind of data have shown great generalization in complex scenes. Videos from YouTube are also used as a data source to train the network [21], which can estimate dense depth maps when both the human and the camera are moving. Lasinger *et al.* [19] proposed a 3D movie dataset and apply joint training of multiple datasets initiated with the structure of [39] to improve the generalization of the model and make it suitable for a variety of scenarios.

2.3 Shallow Depth-of-Field

Bokeh rendering can be applied in the task of autofocus, and it is suitable for images [12] and videos [46]. The existing lens blur methods can be divided into

object space methods and image space methods. Among object space methods [11,26,44], ray tracing methods [26,44] can accurately reproduce the ray integration performed in the camera body. However, this requires large time cost, which is computationally difficult to solve. This has prompted the search for a method to produce blur directly in the image domain. The blur kernel can be produced as a scatter [17] or a cluster [27] operation. There are methods that allow users to control focus parameters and blur amount [2,36]. Wang *et al.* [36] propose a multi-stage model to combine real data and synthetic data to generate a generalized shallow depth of field synthesis effect from a single picture. Wadhwa *et al.* [35] focus on the shallow depth of field effect of the mobile phone, segmenting portrait and the objects on the human body to improve bokeh effect, and dual-pixel (DP) sensors are applied to predict depth maps. A blending scheme of depth-based bokeh rendering [2,5,46] is proposed to generate shallow depth-of-field effects based on composition of images blurred by different kernels. Ignatov *et al.* [13] present a large-scale bokeh dataset and purpose a multi-level network to gradually learn low-level details from lower levels and refines the results on higher resolution.

3 Proposed Method

Rendering bokeh effect from an all-in-focus image is a complicated task which requires obtaining the global information and manipulating the image in low level. The intuitive process usually contains three steps: (i) Monocular depth estimation. (ii) Focal plane detection or saliency detection. (iii) Bokeh rendering on out-of-focus regions. We simplify this process and combine the first two steps into one, which termed as defocus estimation. The defocus map can be written as:

$$D = |d - d_f| \tag{1}$$

where d is the disparity map and d_f denotes the disparity of focal plane. Further, the blur radius R in rendering process can be directly calculated from defocus map:

$$R = K \cdot D \tag{2}$$

where $\cdot$ is used for the entrywise Hadamard product, K is a constant determined by camera parameters. While the real optical rendering in DSLR camera is based on scene radiance [23] instead of image intensity, we propose the radiance module and expect to simulate this transformation. Besides, as we render the image at low resolution, we utilize a simple and efficient strategy to upsample the rendering result and ensure the refocused object is clear. In summary, our model is made up of 4 modules: defocus estimation, radiance, rendering, and upsampling. The pipeline is shown in Fig. 2. In the following, we will describe the details of each module.

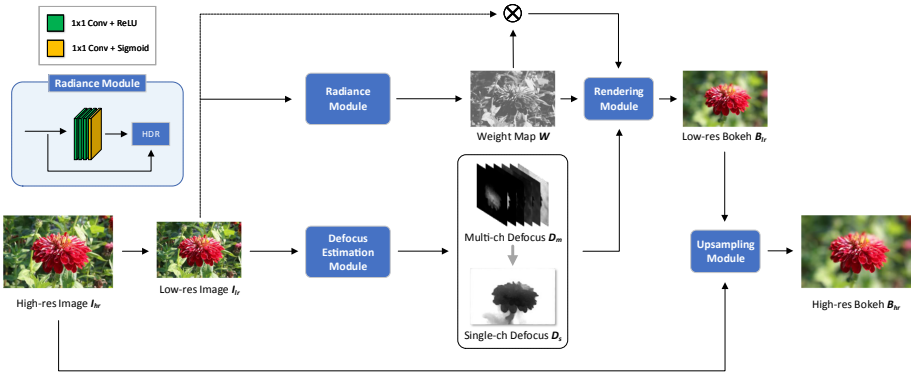

Fig. 2. Pipeline of our proposed method. We first predict defocus map and weight map from the input image of low resolution. Then the bokeh image can be obtained with two predicted maps and pre-defined blur kernels. Finally, we utilize upsampling module to restore the bokeh result to original size.

3.1 Defocus Estimation

Defocus estimation belongs to dense prediction task. Therefore, we use the state-of-the-art U-Net architecture [39] proposed in the field of monocular depth estimation for our defocus estimation network. To accelerate the speed of inference, we utilize ResNeXt50 [41] pre-trained on ImageNet [4] as backbone. In spite of the fact that the architecture performs well in depth estimation, we observe that the network is hard to train when utilizing the single-channel defocus map to render the image directly, so we replace the last single-channel regression layer in original architecture with multi-channel classification layer at training stage 1 and 2. At stage 3, we switch back to the original regression layer. In order to make training process more stable, we also fix most parameters of the pre-trained network and just train the last several layers.

With the defocus estimation network $f_\theta(\cdot)$, the defocus map predicted in different stages can be written as:

$$\begin{aligned} &\text{Stage 1,2:} \quad D_m = f_{\theta_m}(I) \\ &\text{Stage 3:} \quad D_s = f_{\theta_s}(I) \end{aligned} \tag{3}$$

where D_s is the single-channel defocus map in the range of 0 to 1. D_m is the multi-channel defocus map which is normalized by softmax function, so it can also be considered as a probabilistic map. The channels of D_m is set to 6, specifically.

3.2 Radiance

Digital cameras generally have two color rendering strategies: the photofinishing model and the slide or photographic reproduction model. In photofinishing

model, imaging pipeline is different in terms of shooting scenes. The photographic reproduction, using the fixed color rendering, is suitable for most amateur photographers. In this paper, we suppose all of the images are captured in the second mode and the radiance of each pixel only depends on its RGB values. However, as the transformation from image intensity to scene radiance is always nonlinear, it is difficult to establish formulas for forward and inverse transformations. We achieve the similar result by giving pixels different weights in rendering process according to their RGB values. As shown in Fig. 2, we use a simple network to calculate this weight map.

In addition, we observe that the bright pixels whose R, G or B value is very close to the upper bound, i.e., 255 are supposed to have more exaggerated weights as their actual energies are much higher than other pixels in HDR. Therefore, we deal with these pixels separately and give them much larger weights manually.

$$W = M_b \cdot \alpha I^{\beta} + (1 - M_b) \cdot g_\theta(I) \tag{4}$$

where image I is normalized to 0–1. $g_\theta(I)$ generates the initial weight map from input image. M_b is a mask which denotes whether R, G or B value of each pixel is more than a threshold. The threshold is set to 0.99 in this paper. α and β are two hyperparameters. α controls the maximum weight of bright pixels and β adjusts the degree of difference among RGB color channels. We set α to 3 and β to 5. From the experiment, one can see that this operation leads to the HDR effect in final bokeh result.

3.3 Bokeh Rendering

Physically motivated refocusing pipeline proposed in SteReFo [2] is able to mimic the real rendering process efficiently. The core idea is to decompose the scene into different depth layers and composite them back together after blurring each layer with pre-defined blur kernels. We leverage this rendering method and make some appropriate modifications on it. (i) We design a soft disk blur kernel K_s instead of hard disk blur kernel K_h which is discretized by 0 or 1. The formula of two kernels can be described as:

$$K_h\left(x, y, r\right) = \begin{cases} 0 & \left(r^2 - x^2 - y^2 < 0\right) \\ 1 & \left(r^2 - x^2 - y^2 \geq 0\right) \end{cases} \tag{5}$$

$$K_s(x, y, r) = \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{1}{4}(r^2 - x^2 - y^2) + \frac{1}{2}\right) \tag{6}$$

The comparison of two kernels with various kernel sizes are shown in Fig. 3. One can see that the soft kernel looks more like a disk, especially for small kernel size, which will create a more natural CoC effect. (ii) The interval of pre-defined kernel sizes are not set to the same due to the fact that the blur amounts of bokeh images are unbalanced. Specifically, the kernel sizes for different stages are listed in Table 1 where size 1 can be viewed as no blur. (iii) To render more

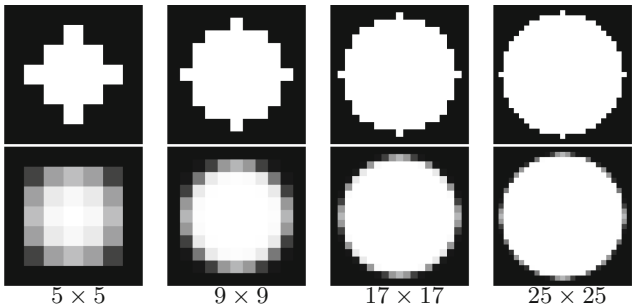

Fig. 3. Visualizations of disk blur kernels with different kernel sizes. The first row is the results of the hard kernel. The second row is the results of the soft kernel designed in this paper.

Table 1. Kernel sizes of different stages.

Stage	1	2	3
Kernel sizes	1, 5, 11, 19, 27, 35	1, 9, 21, 37, 53, 69	1, 3, 5, 7, 11, 15, 19, 23, 27, 33, 39, 45, 53, 61, 69

realistic bokeh effect, we give weights calculated by radiance module to different pixels. For each pixel, the weighted blurring process can be formulated as:

$$B_i = \frac{K(r_i) * (W_i \cdot I_i)}{K(r_i) * W_i} \tag{7}$$

where B_i is the bokeh result of pixel i, W_i are the weights of surrounding pixels I_i within the range of blurring kernel $K(r_i)$. While the blurring operations to each pixel is completely linear, we can extend above process to the whole image.

$$B = \frac{\text{blur\_func}(W \cdot I, D)}{\text{blur\_func}(W, D)} \tag{8}$$

where blur_func is the refocusing pipeline [2] which has been modified as mentioned above. It should be noted that for stage 1 and 2, the defocus map D_m has multi-channel, so there is no need to discretize the defocus map again in rendering process.

3.4 Upsampling

Super-resolution or any other learning based methods [36,37] can be employed to increase the resolution of bokeh result. However, these approaches tend to destroy the CoC effect produced at stage 3. As a result, we calculate a soft mask M_s from predicted single-channel defocus map, and render the final result

B_{hr} by combining the bokeh result B_{lr} upsampled from stage 3 with original all-in-focus image I.

$$B_{hr} = M_s \cdot I + (1 - M_s) \cdot B_{lr} \tag{9}$$

where $M_s = \text{sigmoid}(\eta\,(\tau - D_s))$. η controls the softness degree of transition part between in-focus and out-of-focus regions and τ is a threshold. These two hyperparameters are set to 20 and 1/8, respectively.

3.5 Loss Functions

The network of each stage is end-to-end trainable and we use the same losses for all stages.

$$L_{total} = L_{l_1}(B, B_{gt}) + L_{vgg}(B, B_{gt}) + L_{ssim}(B, B_{gt}) + 0.1 \cdot L_{grad}(D) \tag{10}$$

where L_{l_1} is the L1 loss. L_{vgg} is the perceptual loss which is based on pre-trained VGG19 [16]. L_{ssim} is the structural similarity (SSIM) loss [38]. L_{grad} is the pyramid gradient loss used to constrain the defocus map to be locally smooth, especially in the areas with consistent colors. However, the formula is a little different between the multi-channel defocus map D_m predicted at stage 1, 2 and single-channel defocus map D_s predicted at stage 3.

$$L_{grad}(D_m) = \frac{1}{S}\frac{1}{C}\sum_{i=1}^{S}\sum_{j=1}^{C}\left(\left\|\partial_x D_m^{i,j}\right\|_1 \cdot e^{-\left|\partial_x I^i\right|} + \left\|\partial_y D_m^{i,j}\right\|_1 \cdot e^{-\left|\partial_y I^i\right|}\right) \tag{11}$$

$$L_{grad}(D_s) = \frac{1}{S}\sum_{i=1}^{S}\left(\left\|\partial_x D_m^{i}\right\|_1 \cdot e^{-\left|\partial_x I^i\right|} + \left\|\partial_y D_m^{i}\right\|_1 \cdot e^{-\left|\partial_y I^i\right|}\right) \tag{12}$$

where C is the channel numbers of D_m. S denotes the different scales of the defocus map or image and is set to 4.

4 Experiments

In this section, we first introduce the details about our experimental setting. Then we evaluate the quantitative and qualitative performance of our method on a large-scale bokeh dataset EBB! Finally, we conduct an ablation study to analyse the influence of different factors on the bokeh results.

4.1 Experimental Setup

Our method uses standard PyTorch packages, and is implemented on a single Nvidia GTX 1080 GPU, 251 GB RAM and Intel Xeon Processor. *Everything is Better with Bokeh!* (EBB!) [13] is a large-scale dataset consisting of 4694 aligned wide/shallow depth-of-field image pairs captured using the Canon 7D DSLR with 50 mm f/1.8 lenses. We divide it into a train set with 4224 pairs and a validation set with 470 pairs which is termed as Val470. The training process

is made up of three stages. At stage 1, the model is trained on the resolution of 256 × 256. Initial learning rate used is 1e−4 with a decay-cycle of 30 epochs. At stage 2, the model is fine-tuned on 512 × 512. At stage 3, we replace the multi-channel classification layer with single-channel regression layer in defocus estimation module and train the whole network with most parameters fixed. In addition, for the CPU task of AIM 2020 Rendering Realistic Bokeh Challenge, we only apply the first stage with a residual module for bokeh synthesis.

4.2 Quantitative and Qualitative Evaluation

Quantitative Results. Our model is initially proposed to participate in the AIM 2020 Rendering Realistic Bokeh Challenge [15] and our solution is the runner-up of all methods. The purpose of this challenge is to achieve shallow depth-of-field with the best perceptual quality similar to the ground truth as measured by the Mean Opinion Score (MOS). Table 2 and Table 3 shows the performance of our model. It is worth mentioning that our runtime on GPU is calculated by the speed of TFLite model instead of PyTorch. The average runtime of our PyTorch model on GPU is 0.055 s.

Table 2. Quantitative results of our method from the CPU track of AIM 2020 Rendering Realistic Bokeh Challenge. The results are sorted based on the MOS scores.

Track1: CPU				
Team	MOS↑	PSNR↑	SSIM↑	Avg. runtime (s)
Airia-bokeh	**4.2**	23.58	0.8770	5.52
AIA-Smart	3.8	23.56	0.8829	1.71
CET_SP	3.3	21.91	0.8201	1.17
CET_CVLab	3.2	23.05	0.8591	1.17
Team Horizon	3.2	23.27	0.8818	19.27
IPCV_IITM	2.5	**23.77**	**0.8866**	27.24
CET21_CV	1.3	22.80	0.8628	**0.74**
CET_ECE	1.2	22.85	0.8629	**0.74**

Table 3. Quantitative results of our method from the GPU track of AIM 2020 Rendering Realistic Bokeh Challenge. The results are sorted based on the MOS scores.

Track2: GPU				
Team	MOS↑	PSNR↑	SSIM↑	Runtime (s)
Airia-bokeh	**4.2**	23.58	0.8770	**1.52**
AIA-Smart	3.8	**23.94**	**0.8842**	15.2
CET_CVLab	3.2	23.05	0.8591	2.75

Qualitative Results. In this section, first we present a sample visual result of our solution consisting of intermediate outputs, then we compare our method to the current state-of-the-art solutions [13,14] that were trained and tuned specifically for bokeh rendering. The sample result, which contains a defocus map and a weight map, is shown in Fig. 4, and the visual results of all methods are shown in Fig. 5. As shown in Fig. 5, our method produces a distinctive CoC effect and achieves better quality around the boundaries between in and out-of-focus regions.

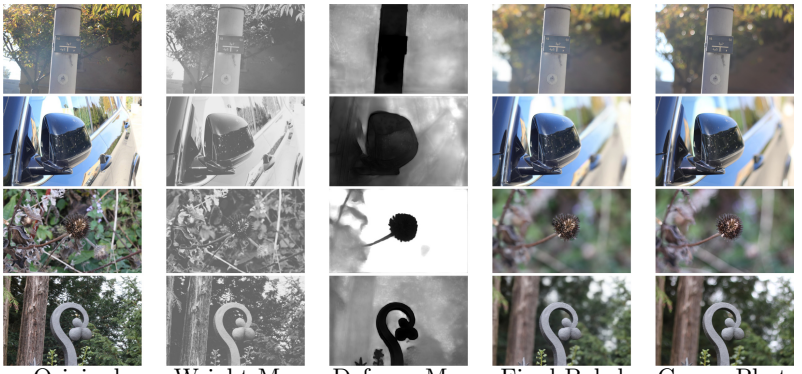

Fig. 4. Visual results obtained with the proposed method. Best zoomed on screen.

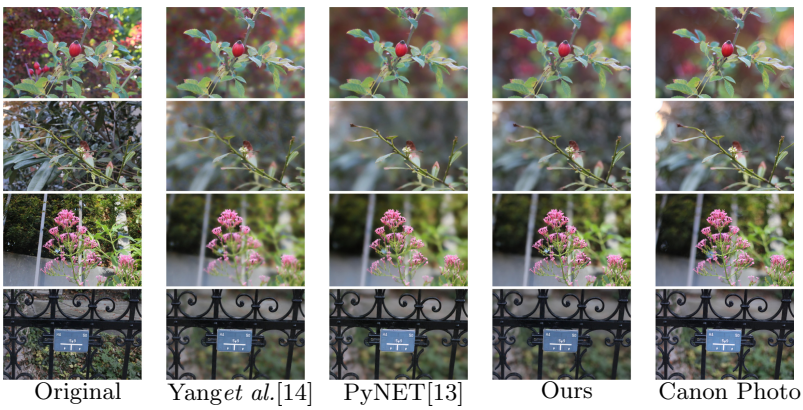

Fig. 5. Visual results obtained with three different methods. From left to right: the original narrow aperture image, Yang *et al.* [14], PyNET [13], our solution and the target Canon photo.

4.3 Ablation Study

As mentioned above, our model consists of different components and is trained in a multi-stage manner with the combination of different losses and various blur kernels. Therefore, we do an ablation study to compare different settings, and demonstrate the effectiveness and superiority of our method. To simplify the verification, we only do the experiments at stage 1 unless otherwise specified.

Combination of Losses. As shown in Table 4, we obtain the best result when all of the losses are used. Besides, we visualize some examples of predicted defocus maps with different combinations of losses in Fig. 6 to verify the effect of each loss. One can see that the predicted defocus maps become more delicate on in-focus regions and smoother on out-of-focus regions by adding loss function gradually. Note that the displayed defocus maps are overlaid by different probabilistic layers produced at stage 1.

Table 4. Quantitative results of stage 1 training with different losses on Val470.

	L_{l1}	$L_{l1} + L_{ssim}$	$L_{l1} + L_{ssim} + L_{vgg}$	L_{total}
PSNR	23.4028	23.4021	23.4306	**23.4495**
SSIM	0.8638	0.8664	0.8665	**0.8668**

Settings of Blur Kernels. The number and size of the blur kernel is largely determined by experience. On the one hand, too many kernels will slow down the running speed while too few kernels will cause the certain limitation. On the other hand, the maximum kernel size is supposed to be consistent with the maximum blur amount in real scene. However, we observe that the defocus map corresponding to the large scale blur is really hard to be learnt as the blur amount varies greatly among the images and most of them are less blurry. In consequence, we do some experiments on the settings of blur kernels for stage 1 and stage 3. we ignore stage 2 because it is similar to stage 1. We compare the different numbers and maximum sizes of blur kernels for stage 1 in Table 5 and Table 6. For stage 3, to generate a smooth bokeh result, the kernel size should be set continuously. We adopt two sampling strategies, i.e., growing sampling and uniform sampling. The growing sampling is the way we mentioned in Sect. 3.3. For uniform sampling, the interval of kernel sizes is set to 4 from beginning to end. The comparison between two strategies is shown in Table 7. The above analysis process can help us set the pre-defined blur kernel better.

Radiance Module. As shown in Table 8, models with radiance module achieve better results. The weight maps obtained from radiance module are able to demonstrate scene radiance, which is more suitable for bokeh rendering than

Fig. 6. Visualization of defocus maps predicted at stage 1. The first row is all-in-focus image. The second row to the last row present the predicted defocus maps training with L_{l1}, $L_{l1}+L_{ssim}$, $L_{l1}+L_{ssim}+L_{vgg}$ and L_{total}, respectively.

Table 5. Quantitative results of stage 1 with different numbers of blur kernels on Val470.

Kernel sizes	1, 7, 19, 35	1, 5, 11, 19, 27, 35	1, 5, 9, 13, 17, 23, 29, 35
PSNR	23.3879	**23.4495**	23.4368
SSIM	0.8647	**0.8668**	**0.8668**

Table 6. Quantitative results of stage 1 with different maximum sizes of blur kernels on Val470.

Kernel sizes	1, 5, 9, 13, 19, 25	1, 5, 11, 19, 27, 35	1, 7, 15, 25, 35, 45
PSNR	23.4156	**23.4495**	23.4004
SSIM	0.8659	**0.8668**	0.8651

Table 7. Quantitative results of stage 3 with different sampling strategies on Val470.

Sampling strategy	Growing	Uniform
PSNR	**23.6776**	23.6757
SSIM	**0.8815**	0.8814

Table 8. Quantitative results of stage 1 with or without radiance module

	w/radiance	w/o radiance
PSNR	**23.4495**	23.2756
SSIM	**0.8668**	0.8649

Fig. 7. Qualitative results of inference stage with and without radiance module.

Table 9. Quantitative results of different stages on Val470.

Stage	1	2	3	Inference
PSNR	23.4495	**23.7480**	23.6776	23.6262
SSIM	0.8668	**0.8821**	0.8815	0.8798
Runtime (s)	**0.030**	0.039	0.054	0.055

image intensity. To further prove this, we show some examples of final inference stage in Fig. 7.

Training Stages. The training process is made up of three stages. At stage 1, predicting multi-channel defocus map at 1/4 resolution is beneficial for the network to learn global features, At stage 2, we predict defocus map at 1/2 resolution to preserve more details around foreground boundaries. To produce more prominent CoC effect, we change the defocus map from multi-channel to single-channel at stage 3. At inference stage, as we render the image at 1/2 resolution, a simple upsampling strategy is utilized to ensure the clarity of the refocused object. We prove these points in Fig. 8. We also list the quantitative result of different stages in Table 9. Although the PSNR and SSIM of stage 3 is slightly lower than those of stage 2, the layer effect of CoC on out-of-focus regions is more pleasing and realistic. In addition, we observe that the indicators decrease again at inference stage while obtaining better visual quality.

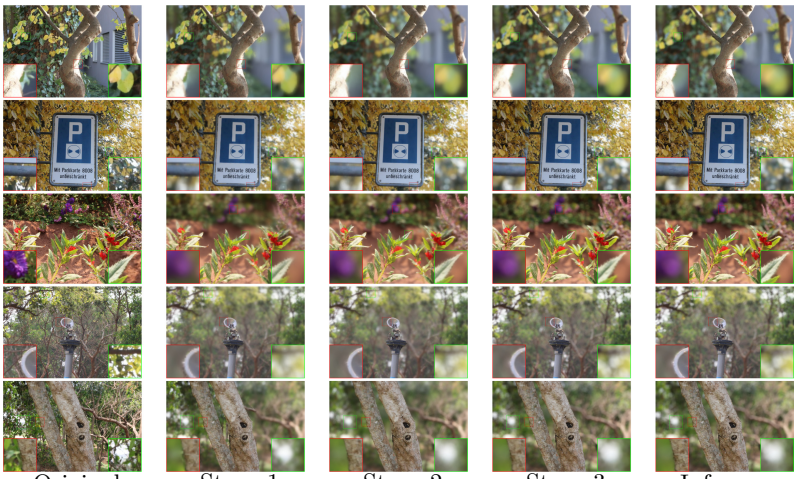

Fig. 8. Qualitatively results of different stages

5 Conclusion

We have presented an effective way to train neural network to predict a shallow depth-of-field image from a single narrow aperture image. By introducing defocus estimation within our network using only bokeh images as supervision, we train our multi-stage network to produce results from multi-channel probabilistic defocus maps to single-channel defocus maps, improving aesthetic quality of synthesized bokeh. Our model also consists of a radiance module which transforms image intensity into scene radiance, rendering bokeh in a physics-based manner. We significantly enhance the quality of bokeh images by applying the above model. Exhaustive visualizations and ablation studies are presented to validate the modules and demonstrate their effects on the performance of our proposed network. In the future we can explore using dynamic filters for selection of kernels and modify the defocus estimation model to learn single-channel defocus maps without trained parameters, eventually decreasing training time.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China (Grant No. U1913602).

References

1. Bae, S., Durand, F.: Defocus magnification. In: Computer Graphics Forum, vol. 26, pp. 571–579. Wiley (2007)
2. Busam, B., Hog, M., McDonagh, S., Slabaugh, G.: SteReFo: efficient image refocusing with stereo vision. In: Proceedings of the IEEE International Conference on Computer Vision Workshops (2019)

3. Chakrabarti, A., Zickler, T., Freeman, W.T.: Analyzing spatially-varying blur. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2512–2519. IEEE (2010)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
5. Dutta, S.: Depth-aware blending of smoothed images for bokeh effect generation. arXiv preprint arXiv:2005.14214 (2020)
6. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658 (2015)
7. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Advances in Neural Information Processing Systems, pp. 2366–2374 (2014)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the KITTI dataset. Int. J. Robot. Res. **32**(11), 1231–1237 (2013)
9. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 270–279 (2017)
10. Guo, X., Li, H., Yi, S., Ren, J., Wang, X.: Learning monocular depth by distilling cross-domain stereo networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 484–500 (2018)
11. Haeberli, P., Akeley, K.: The accumulation buffer: hardware support for high-quality rendering. ACM SIGGRAPH Comput. Graph. **24**(4), 309–318 (1990)
12. Herrmann, C., et al.: Learning to autofocus. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2230–2239 (2020)
13. Ignatov, A., Patel, J., Timofte, R.: Rendering natural camera bokeh effect with deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 418–419 (2020)
14. Ignatov, A., et al.: Aim 2019 challenge on bokeh effect synthesis: methods and results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 3591–3598. IEEE (2019)
15. Ignatov, A., Timofte, R., et al.: AIM 2020 challenge on rendering realistic bokeh. In: Bartoli, A., Fusiello, A. (eds.) ECCV 2020. LNCS, vol. 12537, pp. 213–228. Springer, Cham (2020)
16. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
17. Kriváneк, J., Zara, J., Bouatouch, K.: Fast depth of field rendering with surface splatting. In: 2003 Proceedings Computer Graphics International, pp. 196–201. IEEE (2003)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
19. Lasinger, K., Ranftl, R., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. arXiv preprint arXiv:1907.01341 (2019)
20. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2007)

21. Li, Z., et al.: Learning the depths of moving people by watching frozen people. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4521–4530 (2019)
22. Li, Z., Snavely, N.: MegaDepth: learning single-view depth prediction from internet photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2041–2050 (2018)
23. Lin, H., Kim, S.J., Süsstrunk, S., Brown, M.S.: Revisiting radiometric calibration for color computer vision. In: 2011 International Conference on Computer Vision, pp. 129–136. IEEE (2011)
24. Lin, J., Ji, X., Xu, W., Dai, Q.: Absolute depth estimation from a single defocused image. IEEE Trans. Image Process. **22**(11), 4545–4550 (2013)
25. Park, J., Tai, Y.W., Cho, D., So Kweon, I.: A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1736–1745 (2017)
26. Pharr, M., Jakob, W., Humphreys, G.: Physically Based Rendering: From Theory to Implementation. Morgan Kaufmann, Burlington (2016)
27. Robison, A., Shirley, P.: Image space gathering. In: 2009 Proceedings of the Conference on High Performance Graphics, pp. 91–98 (2009)
28. Shi, J., Tao, X., Xu, L., Jia, J.: Break ames room illusion: depth from general single images. ACM Trans. Graph. (TOG) **34**(6), 1–11 (2015)
29. Shi, J., Xu, L., Jia, J.: Just noticeable defocus blur detection and estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 657–665 (2015)
30. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
32. Srinivasan, P.P., Garg, R., Wadhwa, N., Ng, R., Barron, J.T.: Aperture supervision for monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6393–6401 (2018)
33. Tang, C., Hou, C., Song, Z.: Defocus map estimation from a single image via spectrum contrast. Opt. Lett. **38**(10), 1706–1708 (2013)
34. Tang, C., Wu, J., Hou, Y., Wang, P., Li, W.: A spectral and spatial approach of coarse-to-fine blurred image region detection. IEEE Sig. Process. Lett. **23**(11), 1652–1656 (2016)
35. Wadhwa, N., et al.: Synthetic depth-of-field with a single-camera mobile phone. ACM Trans. Graph. (TOG) **37**(4), 1–13 (2018)
36. Wang, L., et al.: DeepLens: shallow depth of field from a single image. arXiv preprint arXiv:1810.08100 (2018)
37. Wang, X., et al.: ESRGAN: enhanced super-resolution generative adversarial networks. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11133, pp. 63–79. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11021-5_5
38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
39. Xian, K., et al.: Monocular relative depth perception with web stereo data supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 311–320 (2018)

40. Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., Cao, Z.: Structure-guided ranking loss for single image depth prediction. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020
41. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500 (2017)
42. Xu, G., Quan, Y., Ji, H.: Estimating defocus blur via rank of local patches. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5371–5379 (2017)
43. Yan, R., Shao, L.: Blind image blur estimation via deep learning. IEEE Trans. Image Process. **25**(4), 1910–1921 (2016)
44. Yang, Y., Lin, H., Yu, Z., Paris, S., Yu, J.: Virtual DSLR: high quality dynamic depth-of-field synthesis on mobile platforms. Electron. Imaging **2016**(18), 1–9 (2016)
45. Zhang, X., Wang, R., Jiang, X., Wang, W., Gao, W.: Spatially variant defocus blur map estimation and deblurring from a single image. J. Vis. Commun. Image Represent. **35**, 257–264 (2016)
46. Zhang, X., Matzen, K., Nguyen, V., Yao, D., Zhang, Y., Ng, R.: Synthetic defocus and look-ahead autofocus for casual videography. arXiv preprint arXiv:1905.06326 (2019)
47. Zhuo, S., Sim, T.: Defocus map estimation from a single image. Pattern Recogn. **44**(9), 1852–1858 (2011)