Dr.Bokeh: DiffeRentiable Occlusion-aware Bokeh Rendering

Supplementary Material

1. Overview

This supplementary material contains this **pdf** explaining details not covered by the main paper, a **video** explanation and a **html** file for **200** qualitative results.

Sec. 2 discusses our equation details with softened operations. Sec. 3 discusses all the details relevant to differentiability of Dr.Bokeh, including all the derivatives (Sec. 3.1) and more details of depth from defocus using Dr.Bokeh (Sec. 3.2). Sec. 4 talks about how we setup our synthetic benchmark (Sec. 4.1), discussion of the each method defects (Sec. 4.2), more comparison results (Sec. 4.3) and more results of the differentiability of our method (Sec. 4.4).

2. Softened Dr.Bokeh Equation

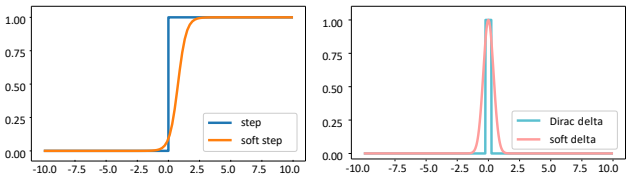

Figure 1. **Softening of the two non-differentiable operations:** Softening of the non-differentiable step function (left). Note we shift the soft-step function to the right to make sure that for $x = 0$, y is close to zero. The right image shows softening of the non-differentiable delta function.

The occlusion-aware bokeh rendering equation in main paper Eqn. 2 includes two non-differentiable terms: the occlusion term O and the scattering term S_l. These terms are non-differentiable as it involves non-differentiable operations similar to step function or Dirac delta function as shown in Fig. 1. We approximate those non-differentiable terms with differentiable operations, e.g., a step function can be approximated with a soft-step function. The occlusion term O is the Dirac delta function δ_x with a value infinity at zero and zero everywhere else. We replace the O_l by:

$$O_l(y,x) = 1 - exp(-3d_x^2)\left(\frac{1}{2}\tanh(10(d_y - d_x - 0.1)) - \frac{1}{2}\right). \tag{1}$$

The scattering term S_l is a step function(if the neighborhood $x + \Delta x$ can scatter to x then is one, otherwise zero) we replace it by a differentiable function:

$$S_l(y,x) = \frac{1}{(1 + 10\exp\left(-3(\alpha|d_y| + 1 - \|d_y - d_x\|_2^2)\right))}, \tag{2}$$

where α is a camera parameter controlling the blur radius. The coefficients in Eqns. (1, 2) are empirically selected to fit the original function and are reasonable to the bokeh rendering process.

3. Differentiable Dr.Bokeh

3.1. All Derivatives

The partial derivative for d is:

$$\frac{\partial L}{\partial B(x)}\frac{\partial B(x)}{\partial d(x)} = \sum_{y\in\Omega}\frac{\partial L}{\partial B(y)}\frac{I(y)(W(y,x) - w(y,x)O(y,x))}{W(y,x)^2} \cdot \left(\frac{\partial w(y,x)}{\partial d(x)}O(y,x) + \frac{\partial O(y,x)}{\partial d(x)}w(y,x)\right). \tag{3}$$

The partial derivative for a is:

$$\frac{\partial L}{\partial B(x)}\frac{\partial B(x)}{\partial a(x)} = \sum_{y\in\Omega(x)}\frac{\partial L}{\partial B(y)}I(y)O(x,y)\frac{\partial w(x,y)}{\partial a(x)} \cdot \frac{W(y) - w(x,y)O(x,y)}{W(y)^2}, \tag{4}$$

where the $W(y)$ is:

$$W(y) = \sum_{y'\in\Omega(y)} w(y',y)O(y',y). \tag{5}$$

The full equation of $\frac{\partial w(x,y)}{\partial d(x)}$ is as follows:

$$\frac{\partial w(x,y)}{\partial d(x)} = \frac{A(x,y)K(x)a(x)\frac{\partial S(x,y)}{\partial d(x)} - S(x,y)K(x)a(x)\frac{\partial A(x)}{\partial d}}{A(x,y)^2} \tag{6}$$

where $\frac{\partial S(x,y)}{\partial d(x)}$ and $\frac{\partial S(x,y)}{\partial d(x)}$ are the following in practice:

$$\frac{\partial S(x,y)}{\partial d(x)} = \frac{0.3e^{3(-(d(x)-d(y)))}}{(e^{3(-(d(x)-d(y)))} + 0.1)^2} \tag{7}$$

$$\frac{\partial O(x,y)}{\partial d(x)} = e^{-3d(x)^2}\frac{-20}{(10(d(y) - d(x) - 0.1))^2} + (-0.5 - \tanh(10(d(y) - d(x) - 0.1)))(-e^{-3d(x)^2}(-6|d(x)|\text{sign}(y))) \tag{8}$$

$$\frac{\partial w(x,y)}{\partial a(x)} = \frac{S(y,x)K(y)}{A(y)} \tag{9}$$

As mentioned in the paper, directly deriving and implementing the backward computation of Dr.Bokeh is complicated so we will release our code.

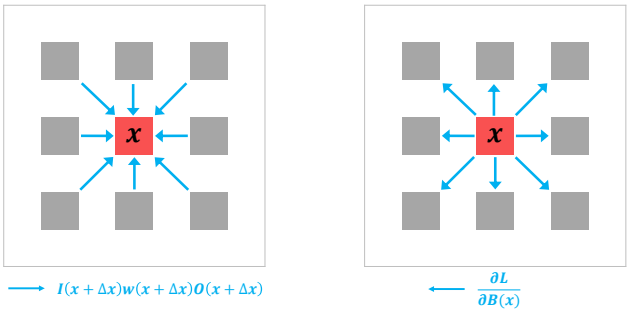

Figure 2. **Per-pixel loss is not enough:** In the left image, neighborhood pixels jointly contribute to x. There exists a case that the x is correct, i.e. the sum of the contributions is correct, but all the neighborhood values are totally wrong, e.g., neighborhood values are shuffled. In this case, there is no loss for pixel x. Then there is no backward gradient for the neighborhood pixels as shown in the right image, even though the neighborhood pixels are wrong.

3.2. Depth from Defocus

We follow existing works [3, 7], the gradient loss in main paper Eqn. 7:

$$G = \frac{1}{N}\sum_{i=1}^{N} |\partial_x D_i| e^{-|\partial_x I_i|} + |\partial_y D_i| e^{-|\partial_y I_i|}, \qquad (10)$$

where N is the N layers of the pyramid of the image, and G is a smooth and regularization term in monocular-depth estimation. In practice, we set $N = 4$.

Except for the gradient loss, we noticed that per-pixel loss, such as $L1$ or $L2$ norm cannot supervise the neural network efficiently due to the ambiguity introduced by the bokeh computation process. As the example shown in Fig. 2, the per-pixel loss fails to supervise the network to optimize the neighborhood values. The reason is that pixel scattering or gathering is a patch-level operation, which means that the per-pixel loss signal is not enough in describing the patch-level error. To guide the network not only care about per-pixel results but also regional results, we propose adding a hierarchy SSIM term to learn a better depth. The default SSIM has a patch size of 11. As the maximum scattering range is highly likely to be larger than 11, we propose to use a hierarchy SSIM loss: a set of SSIM loss with different patch sizes to give the pixel the regional feedback instead of per-pixel feedback.

4. Evaluation

4.1. Bokeh Rendering Evaluation

Existing works [6, 8] setup the scene by compositing multiple layered images and utilizing an approximated pseudo ray tracer to render the lens blur ground truth. Instead, we implemented a renderer that ray traces through a real thin lens to generate the lens blur ground truth in order to evaluate the effectiveness of Dr.Bokeh (see main paper Fig. 6). The lens is modeled as the intersection of two identical spheres of radius R_c, such that the radius of the intersection circle is the aperture radius $R_a = L/2$. The thickness of the lens is computed as $d = 2\sqrt{R_c^2 - R_a^2}$, which gives the lens' focal length f together with the lens' refractive index η, using the lensmaker's equation [2]:

$$\frac{1}{f} = (\eta - 1)\left(\frac{2}{R_c} + \frac{(\eta - 1)d}{\eta R_c^2}\right).$$

The camera is set forth with a chosen FOV, and the image plane is placed at distance $D_I > f$. The color of a pixel is computed by tracing rays from the pixel through various random points on the lens. More details can be found in the appendix.

The scene (5-layer billboards) setup is similar to the dataset by DeepLens [8] and MPIB [6]. The foreground objects are randomly sampled from Adobe Matting Dataset [10] and AIM-500 [4]. The background scenes are randomly sampled from the landmark dataset [9]. The benchmark includes 100 scenes with different blur radiuses and focal planes. Each scene has an all-in-focus image, a ground truth depth, a layered ground truth scene representation, and a bokeh ground truth.

4.2. More Comparisons With Relevant Works

This section discusses the properties of existing methods, analyzes their limitations, and provides more examples of the main artifacts for each method.

SteReFo SteReFo [1] belongs to the classical layered scattering or gathering-based method but with a careful compositing process to "hide" the color bleeding problem. The main issue of the existing methods in this direction suffers from the unnatural boundary partial occlusion effect. See the two main problems for SteReFo in Fig. 3 for examples. The reason for the first problem is that SteReFo does not consider the occluded pixels, which leads to the hard transition on the partial occlusion boundary. The reason for the second problem is that the inter-layer blending weight in SteReFo is not correctly handled.

DeepLens DeepLens [8] proposes a carefully-designed neural network to learn lens blur rendering effectively. One problem is the dataset used by DeepLens is generated by an image space ray tracing method without considering the occluded pixels. The other problem is that the learning process does not naturally preserve the bokeh shape. See Fig. 4

BokehMe BokehMe [5] proposes a hybrid of the classical method and learning-based method to preserve the bokeh shape and also correctly handle the boundary effects by learning from data. BokehMe has done a good job on avoiding the color bleeding problem, but fails to render natural partial occlusion effects. BokehMe tends to render the

Figure 3. **Main problems of SteReFo:** One problem of SteReFo is that the transition boundary for the partial occlusion region is not smooth. The other problem is that SteReFo exaggerates the boundary outward, making the bird in the second row bigger. Best viewed by zoom-in.

Figure 4. **DeepLens problem:** 1. The bokeh shape is not well preserved by DeepLens; 2. The boundary soft transition is not smooth in the partial occlusion boundary. Best viewed by zoom-in.

blurry boundary similar to a glossy window. See Fig. 5 for an example.

MPIB MPIB [6] is the State-of-the-art to handle the color bleeding problem and natural partial occlusion effect rendering simultaneously. MPIB is a variant of multiple plane image (MPI) networks, applies a gathering kernel on the MPI images and learns to blend the different layer results. The common artifact for MPIB is that as it explicitly split the scene into discrete layers, leaking artifacts (also reported by the authors) show up for bad cases. We observe that it

Figure 5. **BokehMe problem:** The transition boundary for BokehMe in partial occlusion is too blurry and lacks soft transition. Best viewed by zoom-in.

Figure 6. **MPIB problem:** Leaking artifacts are consistently shown up due to generalization issue. See the tail of the bird, shoulder of the boy and the ground the boy is standing as an example. Best viewed by zoom-in.

shows up more frequently on unseen data. See Fig. 6 and Fig. 7 as examples.

4.3. More Comparison Results

We provide extensive qualitative results (200 comparative results) in the supplementary(see the supplefig/index.html). In the supplementary, we provide 20 images with different lighting, including day or night, different scenarios, including humans or animals, and different mediums, including photography or paintings. In each case, we show the results of focusing on ten different focal planes. We only show part of the demonstrating results (see Fig. 7) here.

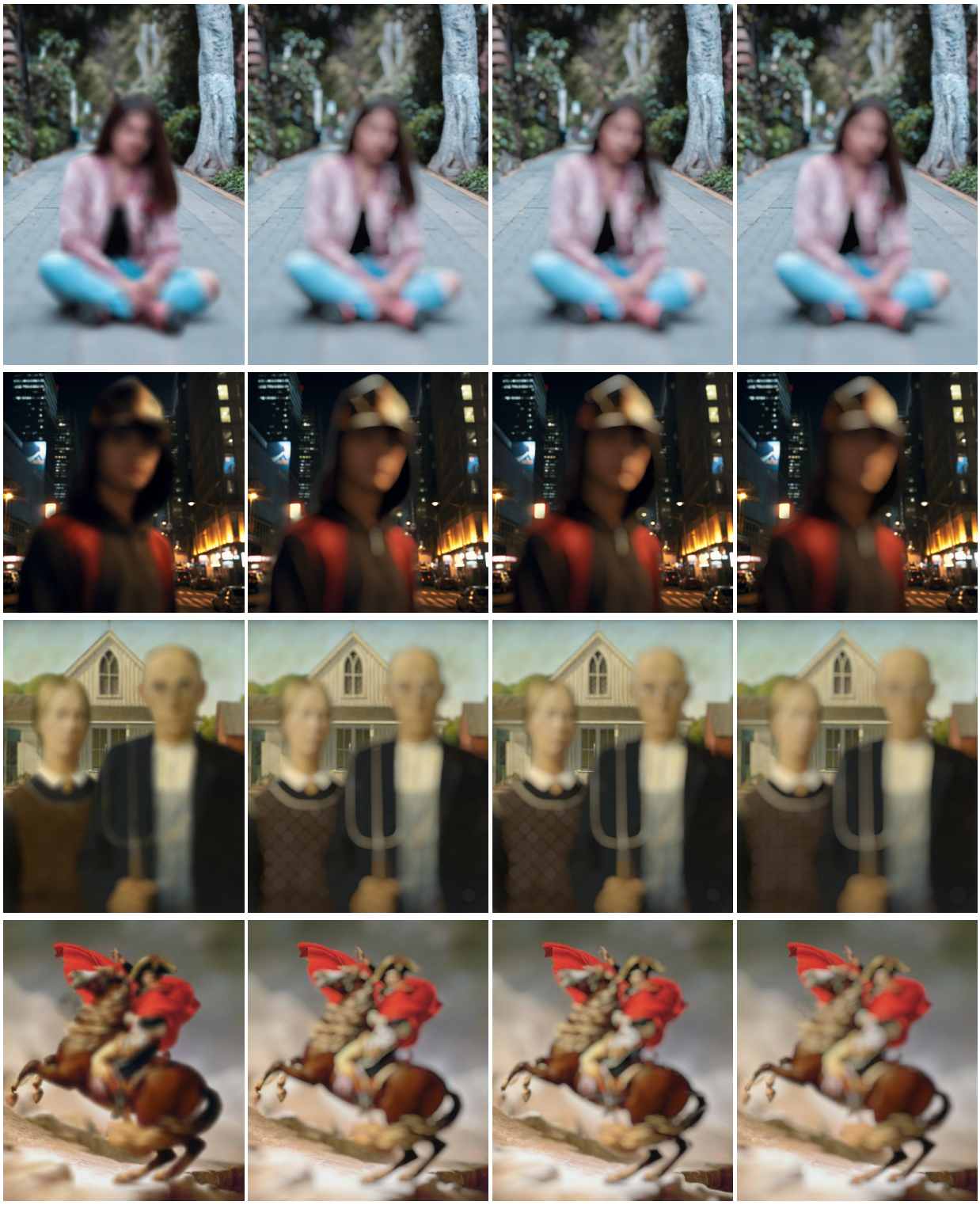

Figure 7. **More demonstrating results.** Best viewed by zoom-in. Please refer to supple-fig.pdf to see all provided results with RGB and depth map inputs.

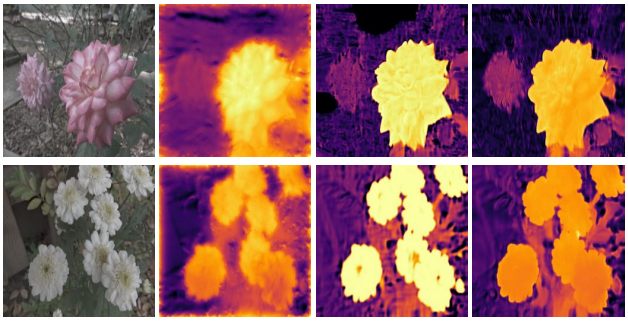

Figure 8. **Depth optimization for one pair data:** The first column is the all-in-focus input image. The second column shows results by Aperture [7]; the third column by GaussPSF [3], and the last column results by Dr.Bokeh. The depth map optimized by Dr.Bokeh has more details and is more accurate.

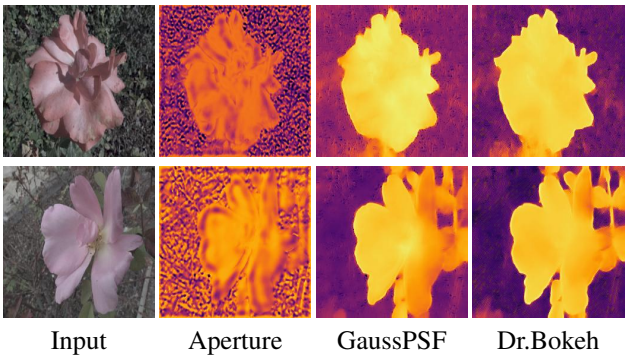

Figure 9. **Depth from defocus dataset**. The first column is the all-in-focus input image. The second column shows results by Aperture [7]; the third column by GaussPSF [3], and the last column by Dr.Bokeh. The depth map by Aperture is noisy. GaussPSF predicts smoother depth.Dr.Bokeh predicts smoother depth and keeps more boundary details.

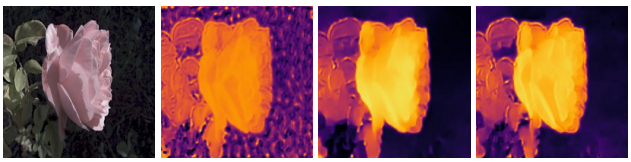

Figure 10. **Qualitative comparisons of different loss:** The first image is the RGB input. The second image is the result of L1 + Grad loss. The third image is the result of L1 + Grad + SSIM loss. The last image is the result of L1 + Grad + HSSIM loss.

4.4. More Differentiability Evaluation

We show the qualitative results of the generated depth map in Fig. 8 and 9. The depth map can either be obtained by direct optimization over an all-in-focus image and a bokeh image pair or by training a neural network to predict the depth based on a large-scale defocus dataset. The direct optimization over one-pair data can clearly show the depth quality supervised by the differentiable rendering layer, while the depth predicted by the trained neural network can illustrate the overall performance of the differentiable layer in the data-driven pipeline.

As shown in Fig. 8, Dr.Bokeh can obtain the best quality depth image supervised by the defocus image as Dr.Bokeh is more accurate in terms of the lens blur physics. Fig. 9 shows that Dr.Bokeh helps the neural networks learn to predict the best quality depth compared with related works.

We provide the qualitative results in our ablation study in Fig. 10. The L1 + Grad loss makes the depth map relatively noisy. The L1 + Grad + SSIM makes the results smoother but loses some details. Our L1 + Grad + HSSIM gets a smooth depth map while preserving the boundary details.

References

[1] Benjamin Busam, Matthieu Hog, Steven McDonagh, and Gregory Slabaugh. SteReFo: Efficient Image Refocusing with Stereo Vision. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3295–3304, Seoul, Korea (South), 2019. IEEE. 2

[2] John E Greivenkamp. *Field guide to geometrical optics*. SPIE press Bellingham, Washington, 2004. 2

[3] Shir Gur and Lior Wolf. Single Image Depth Estimation Trained via Depth From Defocus Cues. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7675–7684, Long Beach, CA, USA, 2019. IEEE. 2, 6

[4] Jizhizi Li, Jing Zhang, and Dacheng Tao. Deep automatic natural image matting. *arXiv preprint arXiv:2107.07235*, 2021. 2

[5] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. BokehMe: When Neural Rendering Meets Classical Rendering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16262–16271, New Orleans, LA, USA, 2022. IEEE. 2

[6] Juewen Peng, Jianming Zhang, Xianrui Luo, Hao Lu, Ke Xian, and Zhiguo Cao. MPIB: An MPI-Based Bokeh Rendering Framework for Realistic Partial Occlusion Effects. In *Computer Vision – ECCV 2022*, pages 590–607, Cham, 2022. Springer Nature Switzerland. 2, 3

[7] Pratul P. Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T. Barron. Aperture Supervision for Monocular Depth Estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6393–6401, Salt Lake City, UT, 2018. IEEE. 2, 6

[8] Lijun Wang, Xiaohui Shen, Jianming Zhang, Oliver Wang, Zhe Lin, Chih-Yao Hsieh, Sarah Kong, and Huchuan Lu. DeepLens: Shallow Depth Of Field From A Single Image. *arXiv:1810.08100 [cs]*, 2018. arXiv: 1810.08100. 2

[9] T. Weyand, A. Araujo, B. Cao, and J. Sim. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In *Proc. CVPR*, 2020. 2

[10] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *Proceedings of the IEEE conference*

on computer vision and pattern recognition, pages 2970–2979, 2017. 2