Supplementary Material
MPIB: An MPI-Based Bokeh Rendering Framework for Realistic Partial Occlusion Effects

Juewen Peng[1], Jianming Zhang[2], Xianrui Luo[1], Hao Lu[1], Ke Xian[1⋆], and Zhiguo Cao[1]

[1] Key Laboratory of Image Processing and Intelligent Control, Ministry of Education, School of AIA, Huazhong University of Science and Technology, China
{juewenpeng,zgcao,xianruiluo,hlu,kexian}@hust.edu.cn
[2] Adobe Research
jianmzha@adobe.com
https://github.com/JuewenPeng/MPIB

This document includes the following contents:

1. Layer compositing formulation for RVR [10] and SteReFo [1].
2. Derivation in the ray-tracing-based bokeh generator.
3. Architecture of AUP-Net.
4. Details of the two proposed metrics: PSNR_{ob} and SSIM_{ob}.
5. Discussion of DeepFocus [9].
6. Details of the user study.
7. Failure Case.
8. More detailed analysis of experiments, including the ablation study on different versions of our framework, and the comparison with the novel view synthesis method MINE [3].
9. More visual results on real-world images, including the comparison with state-of-the-art methods and intermediate results of our approach.

1 Layer Compositing Formulation for Layered Rendering Methods

We summarize and formulate the pipelines of RVR [10] and SteReFo [1] as follows. The formula used in our approach is also displayed for comparison.

$$B_{ours} = \frac{\sum_{i=1}^{N}\left((c_i\alpha_i * K_i)\prod_{j=i+1}^{N}(1-\alpha_j * K_j)\right)}{\sum_{i=1}^{N}\left((\alpha_i * K_i)\prod_{j=i+1}^{N}(1-\alpha_j * K_j)\right)}, \tag{1}$$

$$B_{RVR} = \sum_{i=1}^{N}\left((I\alpha_i^{vis} * K_i)\prod_{j=i+1}^{N}(1-\alpha_j^{vis})\right), \tag{2}$$

⋆ Corresponding author

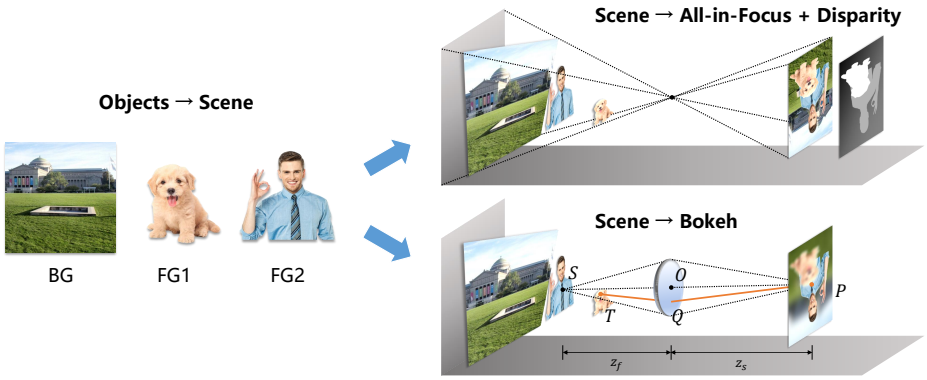

Fig. 1. Pipeline of ray-tracing-based bokeh generator. Given a background RGB image and two foreground RGBA images, we first construct a 3D scene through randomly setting the positions and disparities of different objects. Then, we can synthesize an all-in-focus image and a corresponding disparity map by compositing different planes, and generate a bokeh image by backward tracing the path of light rays.

$$B_{SteReFo} = \frac{\sum_{i=1}^{N}\left((I\alpha_i^{vis} * K_i)\prod_{j=i+1}^{N}(1-\alpha_j^{vis} * K_j)\right)}{\sum_{i=1}^{N}\left((\alpha_i^{vis} * K_i)\prod_{j=i+1}^{N}(1-\alpha_j^{vis} * K_j)\right)}. \tag{3}$$

Both RVR [10] and SteReFo [1] use the whole image I as the images of all layers. Besides, their alpha maps α_i^{vis} are calculated manually and only consider the visible part of the image. For RVR [10], α_i^{vis} is defined by

$$\alpha_i^{vis} = \exp\left(-\lambda\,(D-d_i)^2\right), \tag{4}$$

where D is the disparity map. d_i is the discretized disparity of i-th layer. λ is a hyperparameter to control the transition sharpness of different layers. For SteReFo [1], α_i^{vis} is defined by

$$\alpha_i^{vis} = \frac{1}{2} + \frac{1}{2}\tanh\left(\alpha\,(l-|D-d_i|)\right), \tag{5}$$

where l is the disparity interval of different layers. α is a hyperparameter, which has a similar effect with λ in Eq. 2.

2 Derivation in ray-tracing-based Bokeh Generator

We consider a 3D scene (Fig. 1), where an aperture is at origin, and the image sensor is placed parallel to the aperture with a distance of z_s. For simplification, we assume the image coordinate system and the pixel coordinate system are equivalent.

For each image to be composited, we set its disparity map d as a plane equation of pixel location (x, y):

$$d = \frac{1 - ax - by}{c} \,. \tag{6}$$

Its depth map Z can then be derived as

$$Z = \frac{1}{d} = \frac{c}{1 - ax - by} \,. \tag{7}$$

Considering the perspective projection, when projecting the image to the 3D scene, its x and y coordinates will transform to

$$\begin{cases} X = \dfrac{Z}{z_s} x \,, \\ Y = \dfrac{Z}{z_s} y \,. \end{cases} \tag{8}$$

Then, we can further express Eq. 7 as

$$Z = a z_s X + b z_s Y + c \,. \tag{9}$$

Up to now, we have constructed a 3D scene with several objects in the space. We can use it to easily synthesize an all-in-focus image and a disparity map. The next step is to obtain the rendered result by simulating the propagation of light in the real world. Consider a point $P = (-x, -y, -z_s)$ on the image sensor, we backward trace a ray passing through P and the aperture, and look for the intersection of this ray and the scene. Assume the ray intersects the aperture at point $Q = (u, v, 0)$. According to the triangle similarity theorem, we can calculate the intersection $S = (\frac{z_f}{z_s} x, \frac{z_f}{z_s} y, z_f)$ of the refracted ray and the focal plane, where z_f is the refocused depth. The ray equation from Q to S can be expressed as

$$\begin{cases} X = t\,(\dfrac{z_f}{z_s} x - u) + u \,, \\ Y = t\,(\dfrac{z_f}{z_s} y - v) + v \,. \\ Z = t\, z_f \,, \end{cases} \tag{10}$$

where t is a parameter. Then, we can calculate the intersection T of the ray and each object by simultaneous equations of Eq. 9 and Eq. 10. According to Eq. 8, we can further project T to the sensor plane. Let the projected point be $(-x_n, -y_n, -1)$, we can derive

$$\begin{cases} x_n = x + \dfrac{1 - ax - by - c_n/z_f}{au + bv + c_n/z_s} u \,, \\ y_n = y + \dfrac{1 - ax - by - c_n/z_f}{au + bv + c_n/z_s} v \,. \end{cases} \tag{11}$$

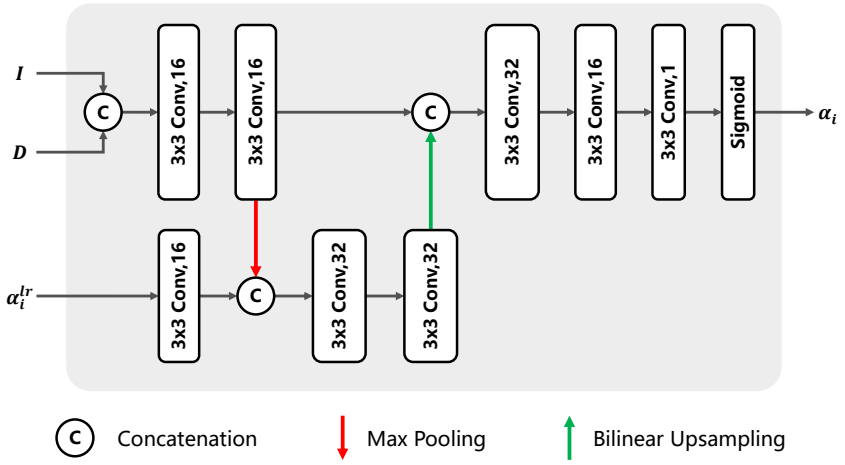

Fig. 2. Architecture of AUP-Net.

Assume the aperture radius is R, we can represent u and v as

$$u = R\mu\,, \quad v = R\nu\,, \tag{12}$$

where

$$\mu^2 + \nu^2 \leq 1\,. \tag{13}$$

Let

$$A = Rz_s\,, \quad d_f = \frac{1}{z_f}\,, \tag{14}$$

where A is the blur parameter, and d_f is the refocused disparity. We can further express Eq. 11 as

$$\begin{cases} x_n = x + \dfrac{1 - ax - by - c_n d_f}{aAu + bAv + c_n} A\mu\,, \\ y_n = y + \dfrac{1 - ax - by - c_n d_f}{aAu + bAv + c_n} A\nu\,. \end{cases} \tag{15}$$

So far, we have obtained Eq. 8 of the main paper.

3 Architecture of AUP-Net

AUP-Net iteratively upsamples the low-resolution alpha map α_i^{lr} of each MPI plane by a factor of 2, which is guided by the high-resolution all-in-focus image I and disparity map D. The network architecture is shown in Fig. 2. Note that except for the last convolution layer, the other convolution layers are followed by a batch normalization layer and a ReLU function.

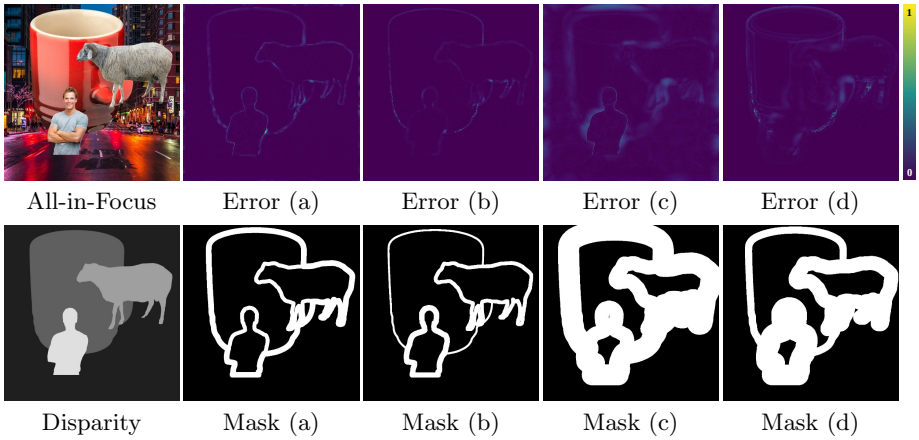

Fig. 3. Error maps of our approach and masks of occluding areas used in the calculation of PSNR_{ob} and SSIM_{ob}. (a) small blur amount, foreground refocusing. (b) small blur amount, background refocusing. (c) large blur amount, foreground refocusing. (d) large blur amount, background refocusing.

4 Details of PSNR_{ob} and SSIM_{ob}

PSNR_{ob} and SSIM_{ob} are used to measure the performance within a mask of occluding boundaries. Specifically, we first produce an initial mask by thresholding the disparity gradient. Then, we pixel-wisely dilate the mask according to the maximum blur radius of surrounding pixels, and PSNR_{ob} and SSIM_{ob} only calculate the corresponding PSNR and SSIM within this mask. We visualize some examples in Fig. 3 for ease of understanding. One can also see that the error of our rendered results is mainly concentrated inside the mask.

5 Discussion of DeepFocus

DeepFocus [9] uses a single neural network to regress bokeh images. Due to the fixed receptive field of the network and the limited blur amount of the training data, DeepFocus can only handle small blur size. If applying large blur size, the network will collapse and produce bokeh images with random colors. Despite the fact that many guided upsampling methods have been proposed, they cannot be directly applied to bokeh rendering, because upsampling in bokeh rendering is much more difficult than other visual tasks. Specifically, to maintain the blur amount of bokeh images unchanged during the upsampling, the blur size is supposed to increase in proportion to the resolution. In addition, as discussed in [2], it is challenging to avoid the blur from upsampling without corrupting the bokeh blur. To verify this, we test several versions of DeepFocus on high-resolution real-world images:

(a) Render the image in original resolution.

Fig. 4. Qualitative results of different versions of DeepFocus [9]. (a) Render the image in original resolution. (b) Render the image in low resolution and bilinearly upsample the rendered result. (c) Render the image in low resolution and upsample the rendered result by guided filter [8]. Results of our approach are displayed in the last column. Disparity maps are predicted by DPT [4].

(b) Downsample the input and decrease the blur parameter before feeding them to the network. Then, bilinearly upsample the rendered result to the original resolution.
(c) As with (b), but upsample the rendered result by guided filter [8].

As shown in Fig. 4, after using the guided filter [8], rendered results in in-focus areas indeed become sharper, but artifacts occur in out-of-focus areas.

6 Details of User Study

We build an online website for user study. The interface of the website is shown in Fig. 5. "Input Image" is an all-in-focus image. "Focal Point" labels the target that are roughly refocused on during the rendering. "Method 1" and "Method 2" display the results of two rendering methods. One is ours. The other is randomly selected from Scatter [6], SteReFo [1], DeepLens [7], DeepFocus [9]. The positions of the two methods are also random. "Magnification Windows" provide simultaneous local zoomed viewings for images in two rows. Users can utilize them to observe and compare the details of the two bokeh images.

7 Failure Case

Our approach has two main limitations: (i) For the object with smoothly varying disparity, it may disconnect at the junction of adjacent planes due to plane discretization. We show an example in the first row of Fig. 6. (ii) The excellent performance of our framework is based on good inpainted results, and the quality

Fig. 5. Interface of the user study website.

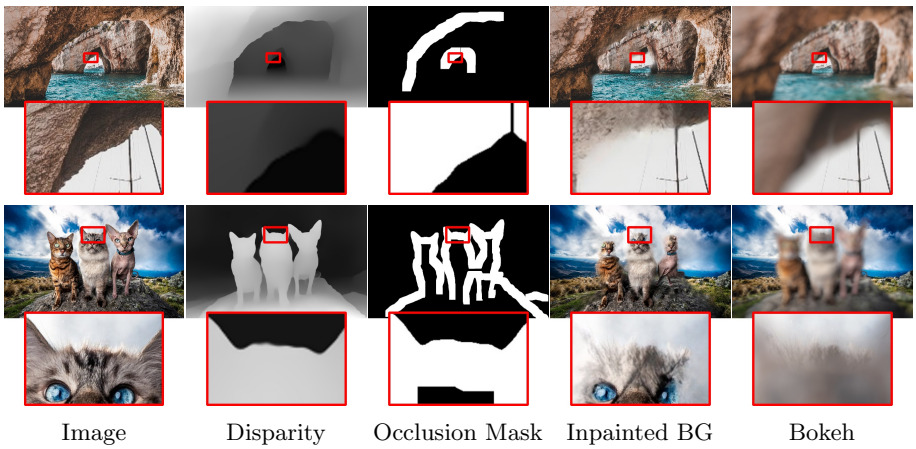

Fig. 6. Failure cases. Row 1: Fracture of the connected area due to plane discretization. Row 2: Boundary artifacts caused by failed inpainting.

of inpainted results depends on the inpainting model and our generated occlusion mask. In the occlusion mask generator, considering the complexity of the actual scene structure and the potential inaccuracy of the input disparity map, we cannot indefinitely extend the occlusion mask until it is closed. Therefore, we will set the fix iterations of extension in advance. In that case, the masked image fed to the inpainting model may contain some foreground information. Typically, the inpainting model fills the mask area according to background textures as we expect. However, it sometimes fails and causes boundary artifacts when the background is refocused on, which can be seen in the second row of Fig. 6.

Table 1. Quantitative results of different versions of our framework on the synthesized dataset. "AuxInpaint" is our final model adopted in the main paper. The best performance is in **boldface**.

Method	Constant disparity for each object					Varying disparity for each object				
	LPIPS↓	PSNR↑	PSNR_{ob}↑	SSIM↑	SSIM_{ob}↑	LPIPS↓	PSNR↑	PSNR_{ob}↑	SSIM↑	SSIM_{ob}↑
OnlyAlpha	0.023	36.4	29.6	0.987	0.942	0.025	36.4	30.2	0.985	0.952
PredRGB	0.020	34.5	27.6	0.985	0.929	0.024	34.9	28.4	0.983	0.938
AuxInpaint	**0.011**	**36.7**	**30.0**	**0.989**	**0.951**	**0.019**	**36.8**	**30.5**	**0.986**	**0.956**

8 More Detailed Analysis of Experiments

8.1 Ablation Study on Framework

MPI representation is a collection of RGBA planes, *i.e.*, $\{(c_i, \alpha_i) \,|\, i = 1, 2, ..., N\}$, where c_i and α_i denote the RGB color and the occupancy of the i-th plane. In the main paper, α_i is explicitly predicted by MPI-Net and upsampled by AUP-Net, while c_i is implicitly obtained by a weighted average of the all-in-focus image I and the inpainted background image I^b, and the blend weight w_i is predicted by MPI-Net. We claim that the inpainting model plays a significant role in rendering realistic partial occlusion effects. To further verify this, we train and test 3 versions of our framework:

(a) OnlyAlpha: As with traditional layered rendering methods, we regard the all-in-focus image as c_i. As a result, the background inpainting module will not be used. In addition, MPI-Net only outputs α_i for each plane.
(b) PredRGB: MPI-Net outputs α_i, w_i and occluded RGB c_i^b for each plane. $c_i = w_i \cdot c_i^b + (1 - w_i) \cdot I$. Similarly, the background inpainting module will not be used. Besides, considering the difficulty of predicting RGB color of each plane, we use ResNet-50 as the encoder of MPI-Net in this setting, while in other settings, we use ResNet-18.
(c) AuxInpaint: Our final model adopted in the main paper. MPI-Net outputs α_i and w_i for each plane. $c_i = w_i \cdot I^b + (1 - w_i) \cdot I$. Note that we only inpaint one background image here.

As shown in Table 1, our final model performs best. It is also interesting to observe that OnlyAlpha performs better than PredRGB in most metrics. However, poor LPIPS of OnlyAlpha may indicate its relatively low perceptual quality. We show this in Fig. 7.

8.2 Comparison with Novel View Synthesis Method.

We compare our model with a novel view synthesis method MINE [3]. Note that for MINE, we use our bokeh rendering module to produce bokeh images from the constructed MPI representation. We visualize some real-world examples in Fig. 8. One can observe that directly applying MINE to bokeh rendering does not work well. There may be several reasons for this: (i) The different characteristics

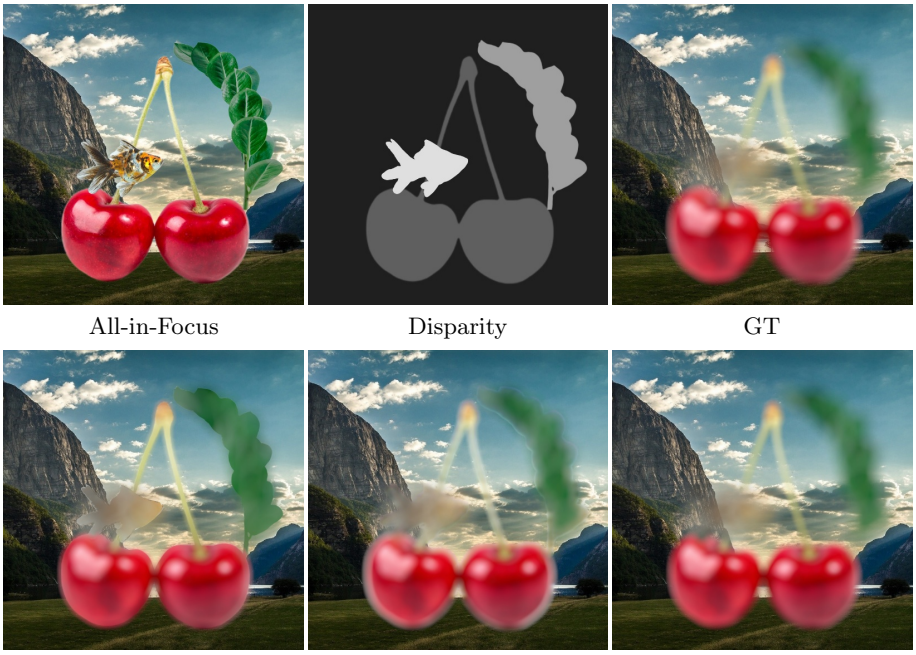

Fig. 7. Qualitative results of different versions of our framework.

of the two tasks. (ii) As we embed the disparity map in the framework, our model can be generalized to a wider range of scenes. (iii) As we combine an off-the-shelf inpainting model and use inpainted results only in occluding areas, our model can easily adapt to high-resolution images.

It is also worth noting that compared with the input disparity map, our reconstructed disparity map D^{rc}, defined by

$$D^{rc} = \sum_{i=1}^{N} \left(d_i \alpha_i \prod_{j=i+1}^{N} (1 - \alpha_j) \right), \tag{16}$$

restores much more detailed and accurate structure information at occluding boundaries, which benefits from AUP-Net. This phenomenon demonstrates that our framework did learn a fine layered scene representation, and explains why our framework is robust to inaccurate disparity inputs and is able to avoid boundary artifacts in most cases.

9 More Visual Results on Real-World Images

9.1 Comparison with State-of-the-Art

We show qualitative comparisons with state-of-the-art methods in Figs. 9 to 11.

9.2 Intermediate Results of MPIB

We show the intermediate results of our approach in Figs. 12 and 13, including the occlusion mask, inpainted background image, and 2 rendered bokeh images refocused on the foreground and the background.

References

1. Busam, B., Hog, M., McDonagh, S., Slabaugh, G.: Sterefo: Efficient image refocusing with stereo vision. In: Proc. IEEE International Conference on Computer Vision Workshops (ICCVW). pp. 0–0 (2019) 1, 2, 6, 12, 13, 14
2. Kim, S.Y., Sim, H., Kim, M.: Koalanet: Blind super-resolution using kernel-oriented adaptive local adjustment. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10611–10620 (2021) 5
3. Li, J., Feng, Z., She, Q., Ding, H., Wang, C., Lee, G.H.: Mine: Towards continuous depth mpi with nerf for novel view synthesis. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 12578–12588 (2021) 1, 8, 11
4. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proc. IEEE International Conference on Computer Vision (ICCV). pp. 12179–12188 (2021) 6, 11, 12, 13, 14
5. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proc. IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 2149–2159 (2022) 15, 16
6. Wadhwa, N., Garg, R., Jacobs, D.E., Feldman, B.E., Kanazawa, N., Carroll, R., Movshovitz-Attias, Y., Barron, J.T., Pritch, Y., Levoy, M.: Synthetic depth-of-field with a single-camera mobile phone. ACM Transactions on Graphics (TOG) **37**(4), 1–13 (2018) 6, 12, 13, 14
7. Wang, L., Shen, X., Zhang, J., Wang, O., Lin, Z., Hsieh, C.Y., Kong, S., Lu, H.: Deeplens: Shallow depth of field from a single image. ACM Transactions on Graphics (TOG) **37**(6), 1–11 (2018) 6, 12, 13, 14
8. Wu, H., Zheng, S., Zhang, J., Huang, K.: Fast end-to-end trainable guided filter. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1838–1847 (2018) 6
9. Xiao, L., Kaplanyan, A., Fix, A., Chapman, M., Lanman, D.: Deepfocus: Learned image synthesis for computational displays. ACM Transactions on Graphics (TOG) **37**(6), 1–13 (2018) 1, 5, 6, 12, 13, 14
10. Zhang, X., Matzen, K., Nguyen, V., Yao, D., Zhang, Y., Ng, R.: Synthetic defocus and look-ahead autofocus for casual videography. ACM Transactions on Graphics (TOG) **38**, 1 – 16 (2019) 1, 2

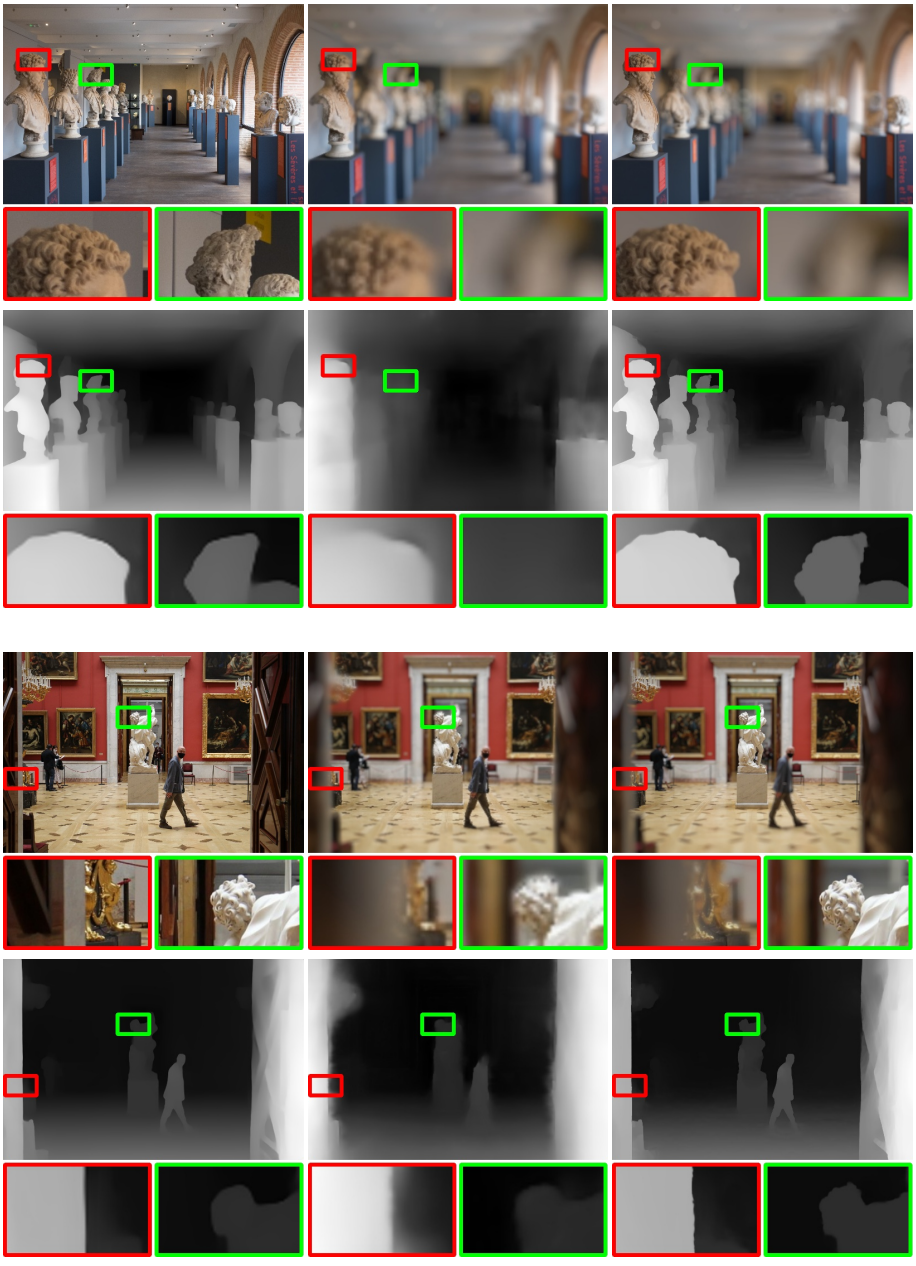

Fig. 8. Comparison with MINE [3]. MINE takes the all-in-focus image as input while our approach takes the all-in-focus image and the disparity map predicted by DPT [4] as input. Column 2 and 3 display the rendered bokeh images and the reconstructed disparity maps of MINE and our approach.

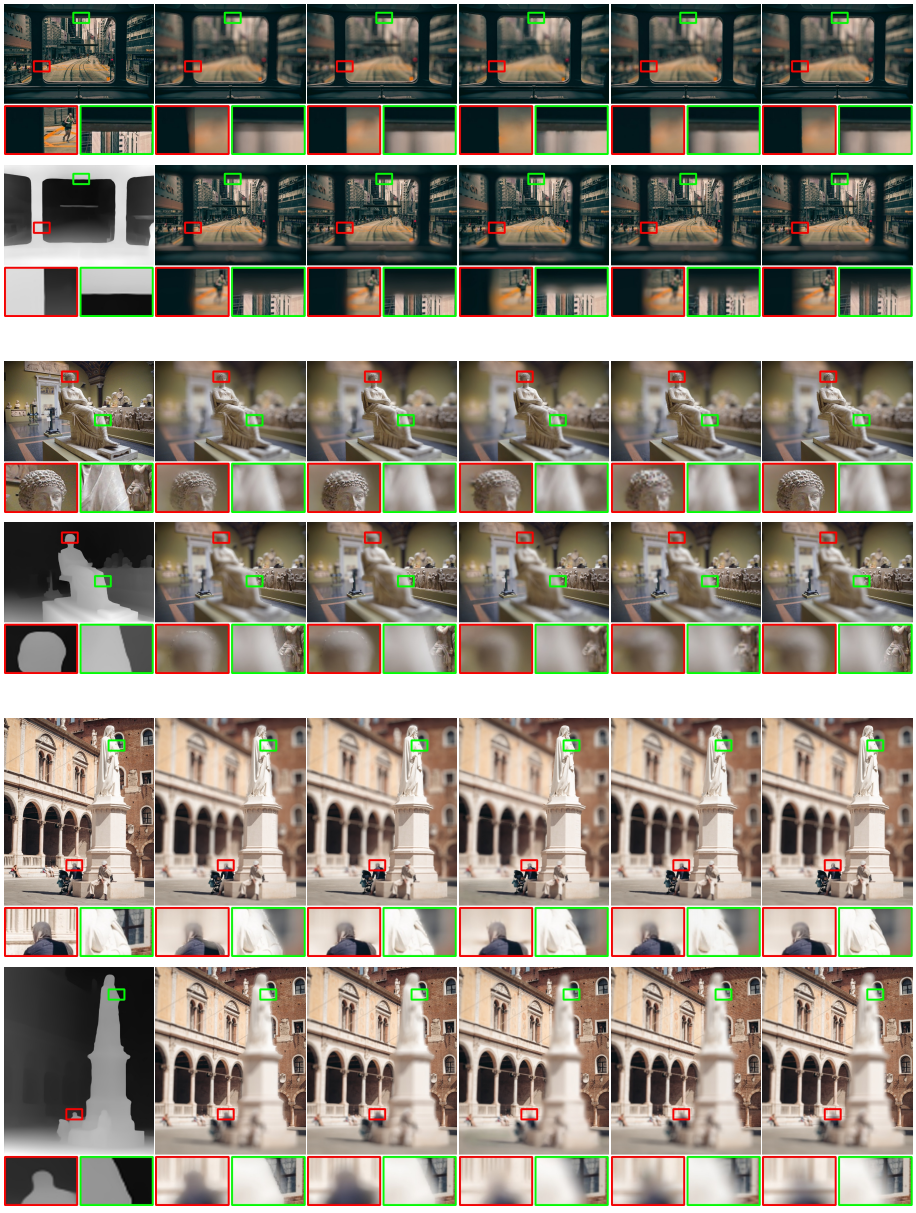

Fig. 9. Qualitative results on real-world images. For each scene, row 1 is refocused on the foreground while row 2 is refocused on the background. Disparity maps are predicted by DPT [4].

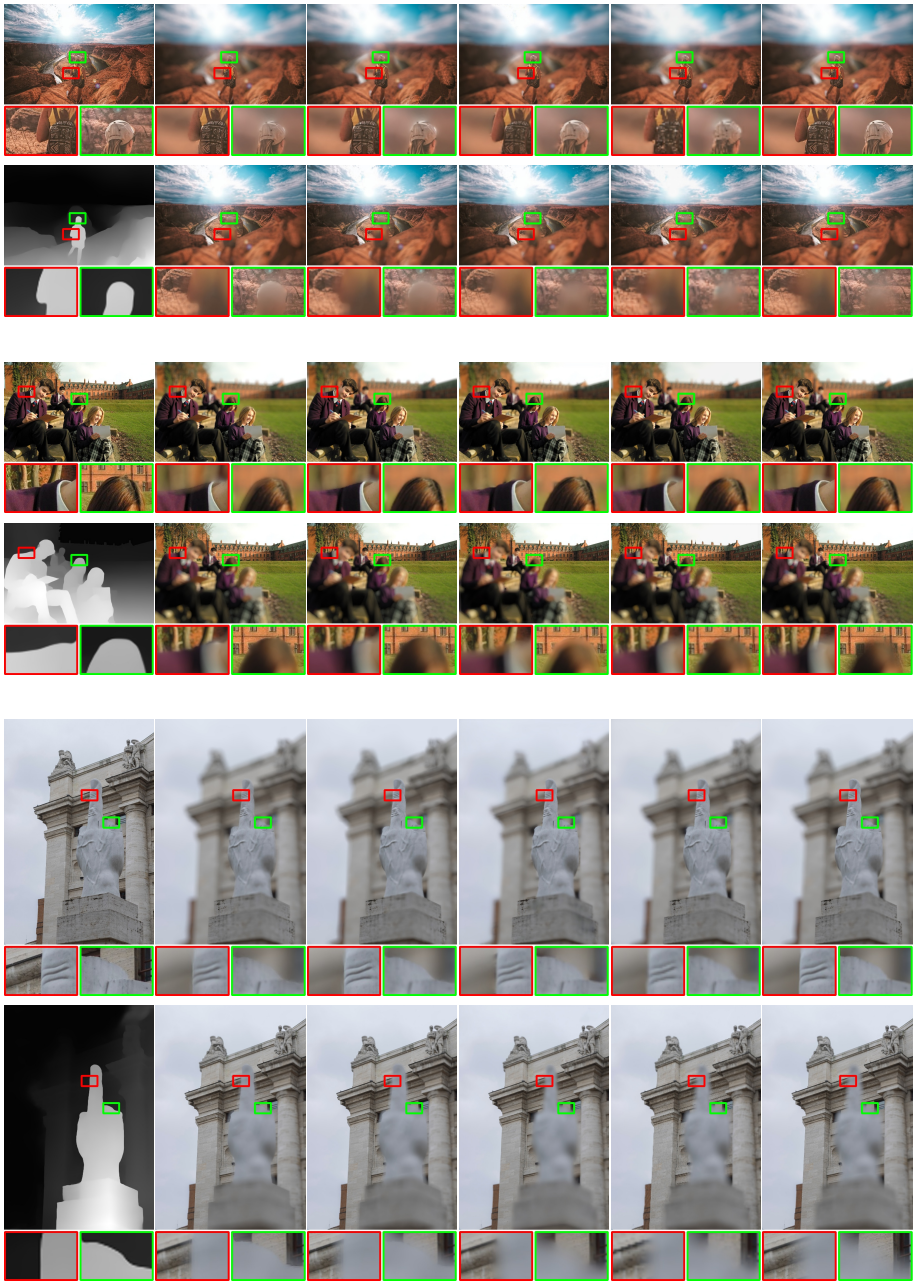

Fig. 10. Qualitative results on real-world images. For each scene, row 1 is refocused on the foreground while row 2 is refocused on the background. Disparity maps are predicted by DPT [4].

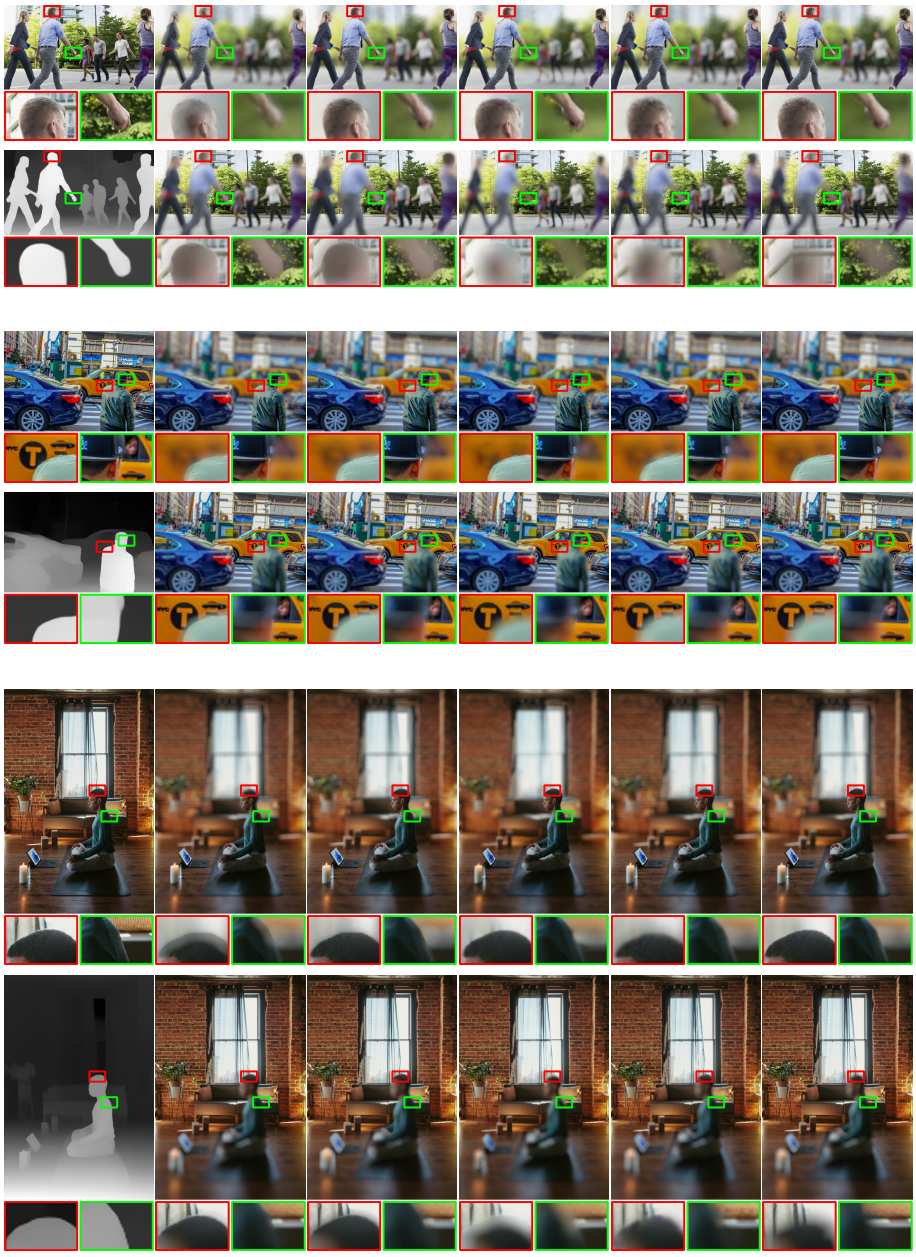

Fig. 11. Qualitative results on real-world images. For each scene, row 1 is refocused on the foreground while row 2 is refocused on the background. Disparity maps are predicted by DPT [4].

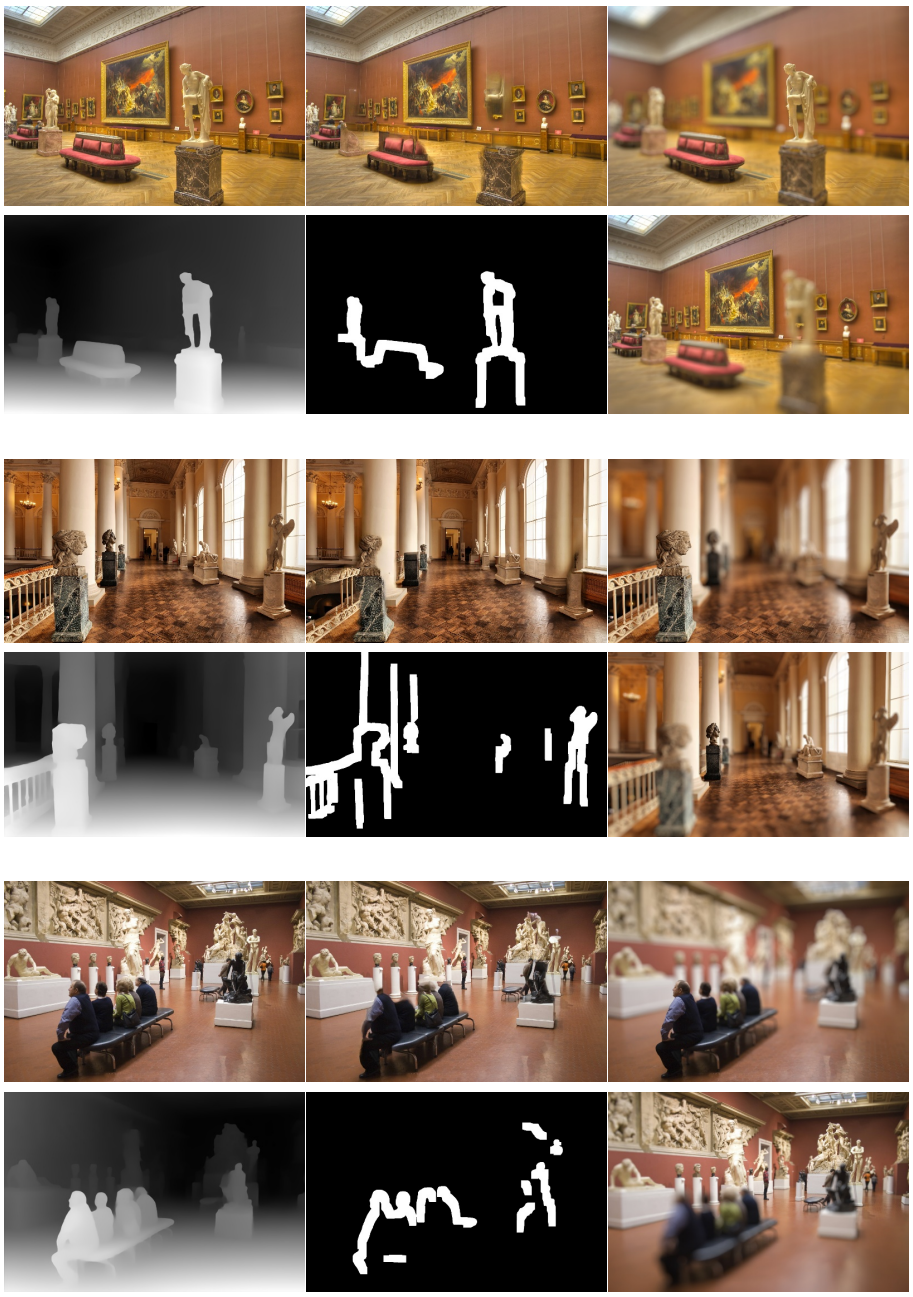

Fig. 12. Intermediate results of our approach. Column 2: Occlusion Mask generated by the occlusion mask generator and background image inpainted by LaMa [5]. Column 3: Rendered bokeh images refocused on the foreground and the background.

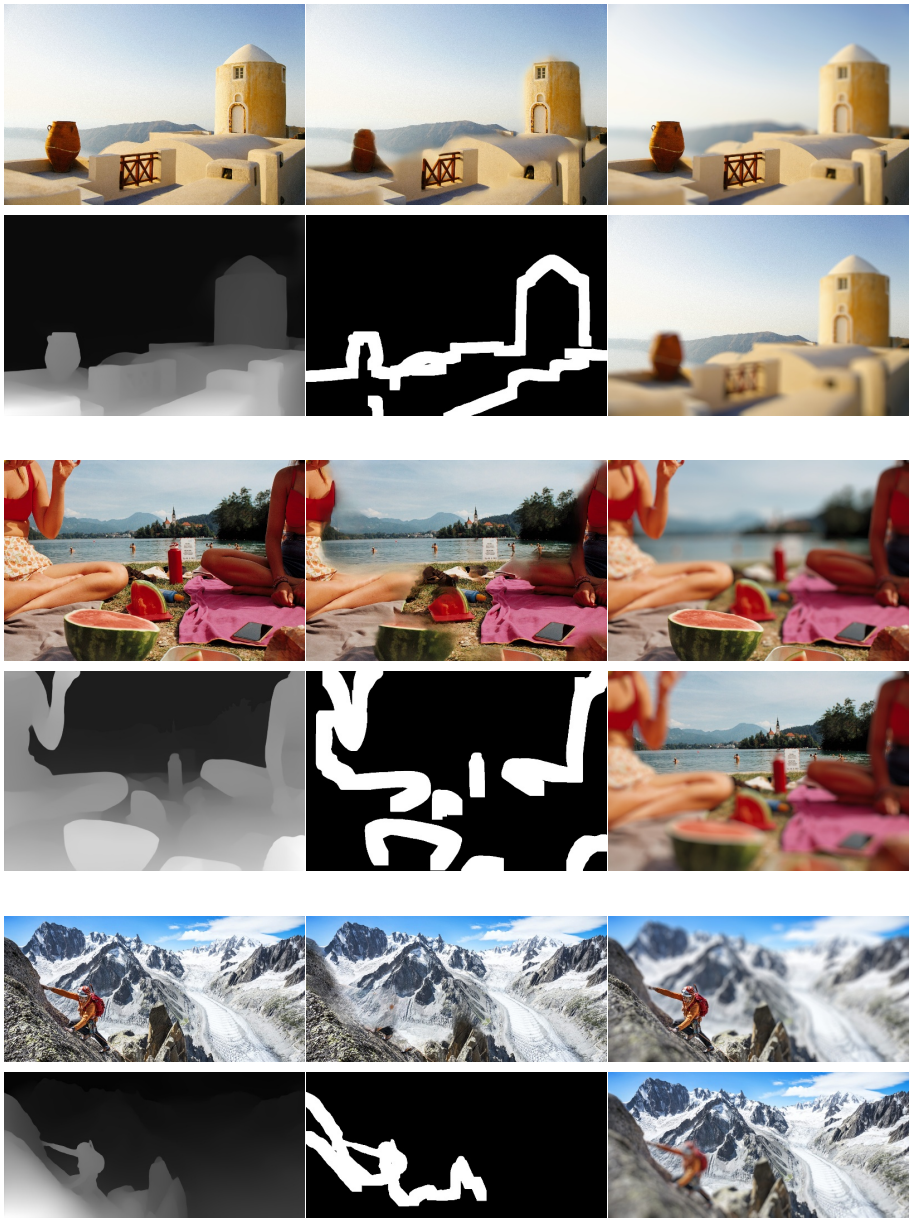

Fig. 13. Intermediate results of our approach. Column 2: Occlusion Mask generated by the occlusion mask generator and background image inpainted by LaMa [5]. Column 3: Rendered bokeh images refocused on the foreground and the background.