Synthetic Depth-Aware Defocus and Controllable Bokeh for Stylized Photography

Jean-Peic Chou
Stanford University
jeanpeic@stanford.edu

Abstract

In this project, we allow users to control the depth of field and the bokeh of photos, which are two main aesthetic features used by photographers and filmmakers. While most existing works consist of end-to-end models generating blur in the background of an image, our model aims at giving users creative freedom by separating the process into different comprehensible steps that we jointly optimize. Even though the trained modules have a limited impact on the resulting images, the overall process shows aesthetically pleasing and realistic results.

Figure 1. Example of swirly bokeh highlighting the central figure.

1. Introduction

Depth of field is a primary aesthetic feature of photography that artists play with to guide the viewer's attention, magnify a subject, or build a scenery. As such, it is essential in our internal representation of an image and our appreciation of it. In practice, the sharpness range and blur quantity depend on the lens aperture, its focal length, the distance to the focus point, and other specific properties of the lens used to capture the scene. Another essential feature of the blur is its bokeh, i.e. the aesthetic quality of the blur of an out-of-focus point of light. Lens constructors meticulously choose the geometry of a lens blades, arrangement of the various optical elements, their size as well as many other elements to create specific bokeh effects. Examples of different shapes of bokeh are shown in Figures 1 and 2. The anamorphic bokeh is for instance key in the Hollywood "cinematic look". Because of the expertise required to combine these elements or the price of the lenses and the camera that can commonly rise to several thousand dollars, such effects are essentially limited to advanced photographers and filmmakers. Additionally, in some cases, physical constraints due to the scene settings or the gears' specificities limit the artistic possibilities, however skillful a photographer can be.

In this work, we present an approach that aims at controlling the focal plane of an image and the strength of the defocus (quantity of blur). In addition, our system enables users to decide the shape and quality of bokeh. With our approach, users can draw the shape of the bokeh and choose if they want the lens used to capture the scene to be anamorphic or not. The system takes a single image as input and generates a defocused image with stylized bokeh based on users' inputs defined earlier. It first predicts a depth map of the input image with the large pretrained version of MiDaS, a vision transformer [21] trained with a mix of multiple datasets [22]. A CNN trained to generate blur then processes this depth map in conjunction with the input image to produce the final result. Our contribution is to propose new controllable features to generate and post-process the blur of an image to help all creators share their own stories through pictures.

2. Related Works

2.1. Synthetic Shallow Depth of Field

There have been many works on rendering synthetic shallow depth of field images. One of the main applications has been to enable mobile phones to take beautiful portraits [20, 26, 24]. In most cases, even when people are not the only focus of these studies, the focusing point remains the most salient part of the image. These models were trained

Figure 2. Example of spherical (regular) and anamorphic bokeh, very common in Hollywood movies.

on large datasets of pairs of aligned images [14, 15, 18], one having a large depth of field, the other with a shallow one. The most recent best-performing methods rely on multi-scale encoder-decoder architectures [12, 11] and can process images very rapidly on CPUs or smartphones to enable most people to benefit from them. However, as end-to-end deep networks, these models do not offer the user controllable features and creative freedom. Neither the focal plane nor the range of the depth of field can be chosen.

Other works try to trace the rays' path in the lens and camera body to obtain accurate physical results [28], but these methods are harder to compute and time costly. To deal with this limitation, some research focus on applying blur kernels with scatter [13] or cluster [23] operations. Many approaches blend images blurred with different strengths to obtain the final result [5]. They are similar to ours as they provide more control [3]. Some works enable users to post-process photos by modifying the camera's parameters as if the picture was being re-shot [19, 9]. But the generated outputs have a generic depth of field and bokeh that cannot be controlled. Even though some works show to be able to produce different types of bokeh [28], none of them propose to fully control it to the best of our knowledge.

2.2. Single Image Depth Estimation

Our work focuses on synthesizing shallow depth of field from a single image as input. In the case of smartphones, many models use additional camera movements during the capture to extract depth information and produce a realistic blur, whether it's parallax information from accidental handshakes [8], multiple lenses [3, 2], or multiple defocus images with a longer capture time [25]. More generally, single image depth estimation is an important research area due to the many useful applications. Recent well-performing methods rely on deep neural networks with end-to-end supervised approaches. Many different multi-level architectures have been proposed following the trends from the deep learning field, from CNNs [7], to VAEs [4] or GANs [1]. Using U-Net architecture to estimate the depth map of images has long been the most common and efficient technique that many works have used for bokeh rendering and many other applications [6, 17]. Today, best performances are achieved by vision transformers [21].

3. Method

Rendering bokeh and modifying the depth of field of an image is a tedious process that requires global and local information. Our method relies on a classical and intuitive pipeline consisting of 3 steps. The separation of these steps enable us to add controls at each of them. A CNN model optimizes the combination of this information to obtain the best results as shown in Figure 3. Once the model is trained, we eventually complete the pipeline with additional user controls.

3.1. Depth Map Prediction

Instead of jointly training a depth estimator with the rest of the model, we use a state-of-the-art vision transformer for depth prediction that was pretrained on more than ten datasets, including photos to 3D movies [22, 21].

Instead of jointly training the transformers model and the rest of the modules which was very time costly and needed more resources, the depth map outputted by the pretrained model is fed into a depth module that assigns for each depth value a size of kernel blur to apply with a non-linear transformation. This method enables to quickly map the depth predicted by the transformers to the blur aspect it should have in average. The module is shown in Figure 4.

3.2. Radiance Estimation

Radiance corresponds to the flux received by an optical system from a certain angle of view. The radiance determines the strength of a bokeh. The more intense a point of light is, the sharper and intense its blurry dot will be. In our case, we suppose that the radiance of each pixel can be determined based on its RGB values. This is approximately the case for photos taken by digital cameras which compute it based on information received by their sensors. There is no exact formula to convert radiance to RGB values or to do the the inverse because this process depends on the camera and the lens. We let a radiance module learn a non-linear dependency between the bokeh intensity at each pixel and the RGB values associated with it thanks to 1x1 convolutional layers as well as tanh and ReLU activations as shown in Figure 5.

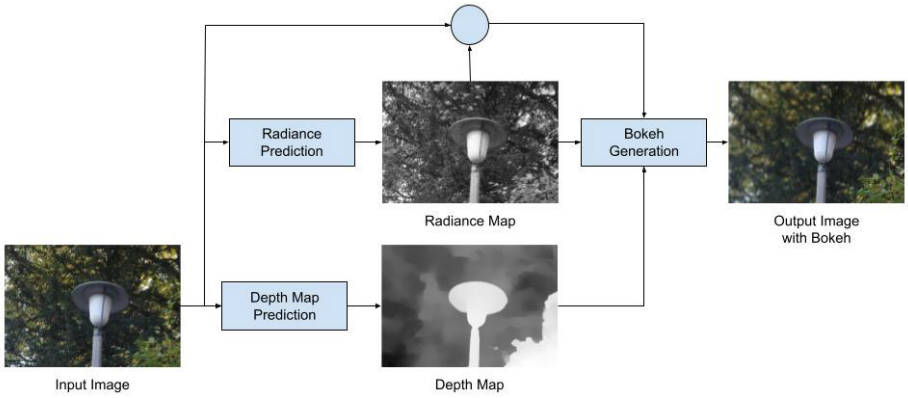

Figure 3. Full model of the bokeh generation process.

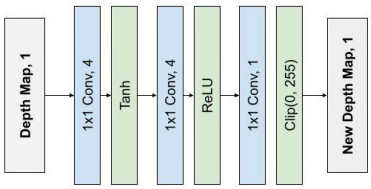

Figure 4. Depth CNN module.

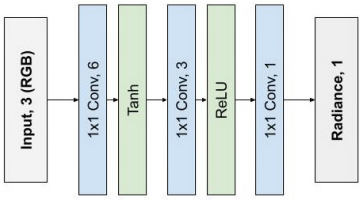

Figure 5. Radiance CNN module.

3.3. Bokeh Generation

The depth map is decomposed into several layers that are blurred with different strengths based on the depth difference with the focal plane chosen by the user. The layers are then composited back together by adding them successively so that the front layers occlude the back ones. During training, we will use soft disk blur kernels proposed by Luo et al. [16] motivated by the physically motivated implementation of SteReFo [3]. They are obtained by applying Gaussian filters. Examples of such kernels are shown in Figure 6. These kernels are applied with weights given by the previously computed radiance, such as:

$$p_i = \frac{\sum_{n \in N_i} K_n * (W_n \cdot p_n)}{\sum_{n \in N_i} K_n * W_n}$$

Where, for pixel p_i, N_i is the set of pixels contributing to the value of p_i, W_n are the radiance weights of the surrounding

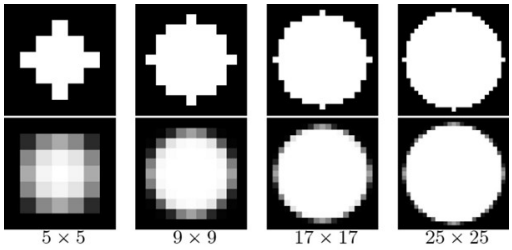

Figure 6. Convolutional blur kernels with a circle shape and different sizes. The second row corresponds to the first one but with soft edges.

pixels p_n interacting via the kernels of weight K_n.

In other words, each pixel creates a dot of light of a size defined by its depth. The strength of the bokeh is determined by its size as well as the radiance of the pixel: disc kernels are normalized to have an $L1$ norm of 1 which means that the larger the disc is, the less it will contribute to an individual neighbor pixel. We compute each pixel's color by adding the contribution of all the surrounding pixels whose bokeh overlaps with it. We then normalize each pixel based on all the contributions to keep the same overall intensity as the original image.

All in all, the model takes as input a single image, and predicts a depth map as well as a radiance map that are used to create the final output based on the process described above. The predicted output is compared to the true image with a shallow depth of field via a sum of the L_1 loss and the structural similarity one defined in [27].

$$L = L_1 + L_{ssim}$$

Backpropagation makes the model learn the radiance module parameters as well as the depth module ones which is equivalent to determining the size of the kernel to apply.

Figure 7. Interface allowing the user to draw a bokeh shape.

3.4. Controls

Once the model is trained for spherical bokeh, we can apply it to other kinds of bokeh. We propose a simple interface shown in Figure 7 that allows a user to draw a bokeh that will be used as resized blur kernels instead of the spherical ones. The drawn shape is blurred with a Gaussian filter in the same way as in the original method. We also add the possibility to change the bokeh into anamorphic by horizontally compressing layers behind the focal plane and vertically doing so for layers in front. The amount of compression is limited to a factor 2 like in most physical lenses and is linear to the depth difference.

We also let the user choose the focal plane as well as the depth of field. The user initially clicks on the part of the image they want to focus on. To keep the depth module results valid, we multiply the size of all the blur kernels by a fixed scaling factor to create shallower or deeper depth of field and thus adjust the blur strength. Eventually, we allow non realistic depth of field effect by allowing the user to define two focus planes, similarly to when photographers and filmmakers use additional half-lenses, and control the depth of field without affecting the bokeh.

4. Dataset

The dataset used in our experiments is Everything is Better with Bokeh! (EBB!) released in 2020 [10]. It contains more than 10 thousand pairs of photos consistently taken in the wild with a Canon 7D DSLR. For each pair, the aperture sizes were chosen at f/16 and f/1.8 to obtain a large depth of field and a shallow one (bokeh effect), respectively. Even though the camera automatically determined the other parameters, we noted slight color, contrast, and exposition differences between corresponding images due to the imperfections of the camera and its lens. This did not affect most end-to-end methods trained to correct these differences while generating the bokeh [12]. As it might have a minor impact on our results, we decided to leave these defaults untouched. Besides, no information is given on the focal length of the lens used to capture these images, but the pictures seem consistent enough to indicate a fixed focal length. The photos were taken during daytime at various moments, i.e. with different lighting environments but never in the dark with high ISO or important grain. The dataset mainly consists of close-up pictures of plants and street objects with a well-defined and clear subject to focus on in the middle of the image. Some example pairs are given in Figure 8. The photos taken by the authors were initially not perfectly aligned. They matched SIFT keypoints with RANSAC to obtain consistent pairs and had to crop some parts of the images. The pictures were downscaled, so that their height was equal to 1024 pixels, but their resulting width was uneven. We decided to crop them all to have a single 1280x1024 pixel format. We also precomputed the depth map of each image and added it to the input. The images were eventually down-scaled by a factor of 2 to accelerate the training process.

We managed to have access to 4,800 pairs of photos from the AIM challenge on rendering realistic bokeh [12]. We divided this dataset into a training dataset containing 4000 images and validation and test sets of 400 pictures each.

5. Evaluation

5.1. Experiments

Each module was designed by varying one hyper parameter and fixing the others while monitoring the validation loss during 1 epoch of training. We tried to use several activation functions and various numbers of filters for each convolutional layers, dropout as well as batch normalization layers. We used Adam optimizer with a learning rate of $1e-2$ after trying various ones ranging from 1 to $1e-5$. The batch size was fixed to 8. We could not go higher due to the lack of memory. One epoch over downscaled images would take about 3 hours.

For the depth map, we tested several state-of-the-art models that aim at predicting it from single images. We empirically kept the model that gave the best results on randomly picked images from the dataset, which was MiDaS [21, 22]. During training, instead of having a user choose a focal plane, we use the closest point determined by the depth map to define the focal plane. It is a general rule of thumb that is most often true for this specific dataset.

Initially, in our method, we successively added layers with occlusion conditions so that deeper layers don't spill on closest ones. By doing so, the borders between each layer were too apparent and unrealistic. After some experi-

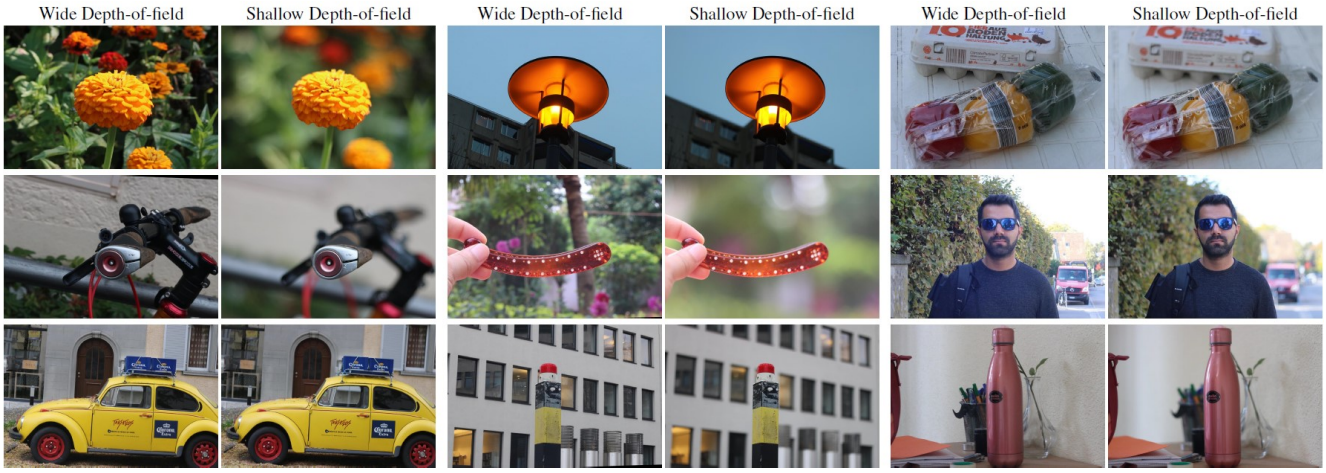

Figure 8. Examples of image pairs from the *Everything is Better with Bokeh!* dataset.

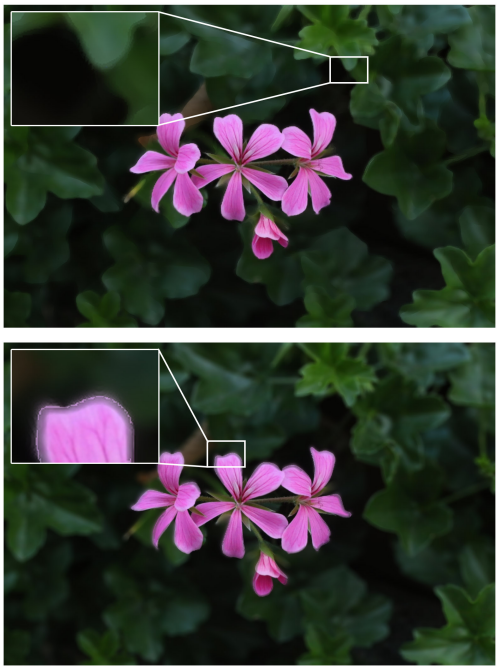

Figure 9. First image is shows full occlusion between layers: The border between 2 layers are sharp cancelling the bokeh effect. Second image shows no occlusion at all: the background bokeh spills on the flower.

mentation, we decided to blend the layers by modifying the opacity of the occluding top layers making the artifacts almost unnoticeable. The two extreme cases of this obstruction issue are presented in Figure 9. In photography, this detail is called the fall-off and is thoroughly examined by experienced artists. This could be another parameter to give access to the users.

5.2. Results Discussion

We compare our results to Stacked DMHN [5] which generates bokeh from a single image by using an end-to-end network composed of two modules of three encode-decoder in cascade. The different levels of the encoders enable the network to leverage local information as well as the global context. It was trained on EBB![12] as well. We tested both models on unseen pictures. We used two metrics which are the ones that define our model's loss: L1 and SSIM. The L1 Score gives us a good approximation of the similarity at a pixel level while SSIM focuses on the similarities of the structural information and the pixel inter-dependencies. The results are shown in Table 1. The baseline outperforms our model for both metrics. This seems reasonably logical since our system was built to be favor control rather than strict performance. One example of our model's advantage is that when no "salient part" is detected like in landscapes, DMSHN is unable to correctly transform the input [5] and leaves it unchanged. In contrast, our method lets the user control the focal plane as shown in Figure 10.

	DMSHN	Ours
L1	**8.33**	12.18
SSIM	**0.86**	0.83

Table 1. L1 and SSIM scores of DMSHN [5] and ours

While training, the validation loss barely decreases indicating that the model cannot generate accurate results with this method. We independently studied each module to observe how they transform the RGB values into radiance and the depth map into an adapted one. The radiance module effect is rather insignificant giving to almost all the input RGB values the same radiance value, i.e. the same weight contribution to the final bokeh. The depth module doesn't

Figure 10. Our system predicts a satisfying depth map for this landscape of Stanford. A focal plane can therefore be chosen to generate bokeh. Here, the focus is on the background buildings making the foreground trees blurred.

change the depth at all as it is unable to map the size of the bokeh to the predicted depth output of the pretrained transformer.

Because our model can not adapt itself to the predicted depth map, its results crucially relies on it. In Figure 11, the baseline's defocus is more convincing while our approach is really dependent on the predicted depth map. In the first row, we observe that because the table is detected in the foreground, it automatically chose the focal plane too close making the central bottle blurry. In the other rows, all layers are well-segmented but the value of the depth is not correctly interpreted, resulting in a soft bokeh compared to the original. Of course, we could individually re-scale the depth. To solve that issue, the depth map could also be predicted jointly with the rest of the parameters as initially intended. However, the size of the bokeh depends on the closeness to the focus plane and other parameters inaccessible by the model. For this reason, the task is actually imperfect and should also mainly be analyzed through an aesthetic lens. Except for the bottle, the images remain rather natural-looking and pleasing to the eye.

Overall, the strict performance of our model was, again, not the point of this work. In Figure 12, we show how this system can be used to generate aesthetic bokeh chosen by a user that can't be achieved by most other models including the baseline. Common types of bokeh used by artists are not circular but donut-shaped or result from an anamorphic lens. These looks bring new layers to the story conveyed by photos that still have to be explored.

6. Conclusion

In this work, we decomposed the bokeh generation process into modules that we jointly trained. This enabled us to control each step and to let users decide of the depth of field features as well as the bokeh ones. Although our model's score performance is worse than the one of the baseline, the aesthetic quality of our results is still very satisfying. A specific look at the trained modules showed that their impacts were minor or nonexistent for the depth one. With more time and more resources, the depth module could be replaced by the depth prediction pretrained model that would be finetuned to produce a coherent depth map. We did not have enough time to implement more features such as the swirly bokeh or a full interface which was not expected for this project. But in the future and thanks to this project, we hope that a bokeh generation tool will be developed and made available to the public, blazing new trails in tomorrow's photography and filmmaking.

References

[1] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In L. Leal-Taixé and S. Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 337–354, Cham, 2019. Springer International Publishing.

[2] J. T. Barron, A. Adams, Y. Shih, and C. Hernández. Fast bilateral-space stereo for synthetic defocus. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4466–4474, 2015.

[3] B. Busam, M. Hog, S. McDonagh, and G. Slabaugh. Sterefo: Efficient image refocusing with stereo vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[4] P. Chakravarty, P. Narayanan, and T. Roussel. Gen-slam: Generative modeling for monocular simultaneous localization and mapping, 2019.

[5] S. Dutta, S. D. Das, N. A. Shah, and A. K. Tiwari. Stacked deep multi-scale hierarchical network for fast bokeh effect rendering from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2021.

[6] F. J. Galetto and G. Deng. Single image deep defocus estimation and its applications, 2021.

[7] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue, 2016.

[8] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. S. Kweon. High-quality depth from uncalibrated small motion clip. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5413–5421, 2016.

[9] B. Huhle, T. Schairer, P. Jenke, and W. Straßer. Realistic depth blur for images with range data. In A. Kolb and R. Koch, editors, *Dynamic 3D Imaging*, pages 84–95, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

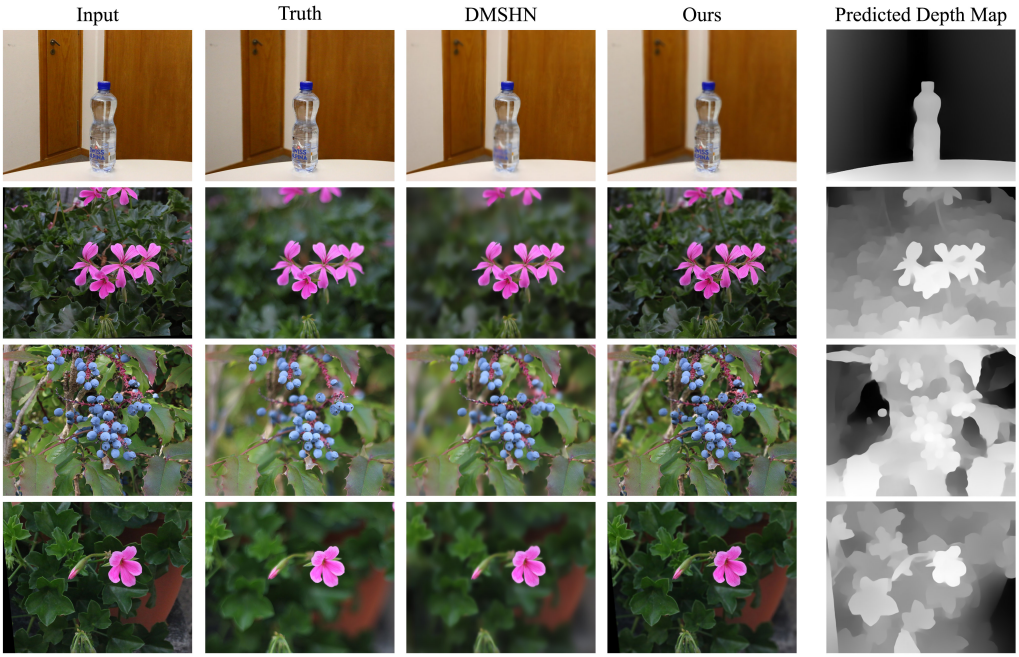

Figure 11. Results on a few examples compared with the input, the truth and our baseline. The predicted depth map taken as input of our model is also shown on the right.

Figure 12. Top left corner: Input image. Top right: Soft spherical bokeh. Bottom left: Donut bokeh. Bottom right: Anamorphic bokeh.

[10] A. Ignatov, J. Patel, and R. Timofte. Rendering natural camera bokeh effect with deep learning, 2020.

[11] A. Ignatov, J. Patel, R. Timofte, B. Zheng, X. Ye, L. Huang, X. Tian, S. Dutta, K. Purohit, P. Kandula, M. Suin, A. N. Rajagopalan, Z. Xiong, J. Huang, G. Dong, M. Yao, D. Liu, M. Hong, W. Lin, Y. Qu, J.-S. Choi, W. Park, M. Kim, R. Liu, X. Mao, C. Yang, Q. Yan, W. Sun, J. Fang, M. Shang, F. Gao, S. Ghosh, P. K. Sharma, A. Sur, and W. Yang. Aim 2019 challenge on bokeh effect synthesis: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3591–3598, 2019.

[12] A. Ignatov, R. Timofte, M. Qian, C. Qiao, J. Lin, Z. Guo, C. Li, C. Leng, J. Cheng, J. Peng, X. Luo, K. Xian, Z. Wu, Z. Cao, D. Puthussery, J. C, H. P. S, M. Kuriakose, S. Dutta, S. D. Das, N. A. Shah, K. Purohit, P. Kandula, M. Suin, A. N. Rajagopalan, S. M. B, M. A. L, S. A. R, P. S, G. Wu, X. Chen, T. Wang, M. Zheng, H. Wong, and J. Zou. Aim 2020 challenge on rendering realistic bokeh, 2020.

[13] J. Krivanek, J. Zara, and K. Bouatouch. Fast depth of field rendering with surface splatting. In *Proceedings Computer Graphics International 2003*, pages 196–201, 2003.

[14] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[15] W. Lijun, S. Xiaohui, Z. Jianming, W. Oliver, L. Zhe, H. Chih-Yao, K. Sarah, and L. Huchuan. Deeplens: Shallow depth of field from a single image. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):6:1–6:11, 2018.

[16] X. Luo, J. Peng, K. Xian, Z. Wu, and Z. Cao. Bokeh rendering from defocus estimation. In A. Bartoli and A. Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 245–261, Cham, 2020. Springer International Publishing.

[17] A. Mertan, D. J. Duff, and G. Unal. Single image depth estimation: An overview. *Digital Signal Processing*, 123:103441, 2022.

[18] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[19] H. Ouyang, Z. Shi, C. Lei, K. L. Law, and Q. Chen. Neural camera simulators, 2021.

[20] J. Peng, X. Luo, K. Xian, and Z. Cao. Interactive portrait bokeh rendering system. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2923–2927, 2021.

[21] R. Ranftl, A. Bochkovskiy, and V. Koltun. Vision transformers for dense prediction. *ArXiv preprint*, 2021.

[22] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[23] A. Robison and P. Shirley. Image space gathering. In *Proceedings of the Conference on High Performance Graphics 2009*, HPG '09, page 91–98, New York, NY, USA, 2009. Association for Computing Machinery.

[24] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs. Automatic portrait segmentation for image stylization. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics*, EG '16, page 93–102, Goslar, DEU, 2016. Eurographics Association.

[25] S. Suwajanakorn, C. Hernandez, and S. M. Seitz. Depth from focus with your mobile phone. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3497–3506, 2015.

[26] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics*, 37(4):1–13, aug 2018.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[28] Y. Yang, H. Lin, Z. Yu, S. Paris, and J. Yu. Virtual dslr: High quality dynamic depth-of-field synthesis on mobile platforms. In *Digital Photography and Mobile Imaging*, 2016.