Selective Bokeh Effect Transformation

Juewen Peng[1], Zhiyu Pan[1], Chengxin Liu[1], Xianrui Luo[1], Huiqiang Sun[1], Liao Shen[1],
Ke Xian[2], and Zhiguo Cao[1*]
[1]Key Laboratory of Image Processing and Intelligent Control, Ministry of Education,
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]S-Lab, Nanyang Technological University
{juewenpeng, zhiyupan, cx_liu, xianruiluo, shq1031, leoshen, zgcao}@hust.edu.cn
ke.xian@ntu.edu.sg
https://github.com/JuewenPeng/SBTNet

Abstract

Bokeh effect transformation is a novel task in computer vision and computational photography. It aims to convert bokeh effects from one camera lens to another. To this end, we introduce a new concept of blur ratio, which represents the ratio of the blur amount of a target image to that of a source image, and propose a novel framework SBTNet based on this concept. For cat-eye simulation and lens type transformation, a two-channel coordinate map and a two-channel one-hot map are added as extra inputs. The core of the framework is a sequence of parallel FeaNets, along with a feature selection and integration strategy, which aims to transform the blur amount with arbitrary blur ratio. The effectiveness of the proposed framework is demonstrated through extensive experiments, and our solution has achieved the top LPIPS metric in NTIRE 2023 Bokeh Effect Transformation Challenge.

Canon50mmf1.4 → Sony50mmf16.0

Canon50mmf16.0 → Sony50mmf1.4

Figure 1. Bokeh effect transformation results of our method. Images are from dataset BETD [4].

1. Introduction

The objective of mobile camera technology is to enhance the visual quality of images and approach the level of professional full-frame cameras. Recent years have seen remarkable advancements in creating realistic and aesthetic bokeh effects from sharp images and deblurring blurred images to restore the missing information. Currently, however, no methods have tried to convert the bokeh effect from one lens to that of another lens. Specifically, given a single image, we need to generate a target image with specific bokeh effect according to information such as lens type, focal length and aperture size in terms of the source lens and the target lens (Fig. 1). This task combines the characteristics of bokeh rendering and defocus deblurring, and is more flexible and more challenging.

With regard to the bokeh rendering, it can be divided into automatic bokeh rendering [5, 8–11, 17, 20] and controllable bokeh rendering [3, 18, 19, 25, 29, 32]. The former one requires a single image input and automatic focusing and blurring. The latter one, on the other hand, enables extra inputs, including a disparity map and some controlling parameters such as the blur amount and the refocused disparity, and the output should be adjusted by these parameters. In any case, however, the input image is required to be all-in-focus or has a deep depth of field. Defocus deblurring, on the contrary, produces a sharp image from a shallow depth-of-field image. For both tasks, the transformation of the blur amount is definitely unidirectional and there is no intermediate state, *e.g.*, from a blurred image to a more blurred image or the opposite one. To process

*Corresponding author.

this novel bokeh effect transformation task, NTIRE 2023 Bokeh Effect Transformation Challenge [4] is hold, and a large-scale corresponding dataset BETD is introduced.

In this paper, we propose a new framework termed SBTNet to specialize in this task. We first apply AlphaNet to extract an alpha map of the focused object, which benefits the maintaining of the sharp foreground regions in the transformed result. To simulate the cat-eye effect of bokeh images captured by real lens, we add a two-channel coordinate map as an extra input. A two-channel one-hot map is also added for the transformation of the lens type. To perform the transformation of blur amount, we introduce a new concept of blur ratio, which indicates the ratio of the blur amount of a target image to that of a source image. Based on this concept, we apply several parallel FeaNets and design a feature selection and integration strategy to transform the bokeh effect of images with arbitrary blur ratio. We further use RefineNet to refine the transformed result and restore it to the full resolution. Extensive experiments show that SBTNet can render realistic transformation effect for different blur ratios, and our solution has obtained the top LPIPS metric in the NTIRE 2023 Bokeh Effect Transformation Challenge [4].

Our main contributions are summarized as:

- We define a new concept of blur ratio to model the blur amount transformation and unite the bokeh rendering and defocus deblurring tasks.
- We propose a novel framework with a feature selection and integration strategy to tackle the bokeh effect transformation task with arbitrary blur ratio.

2. Related Work

Bokeh effect transformation is a brand new task, which aims to transform the bokeh style from one lens to another. The bokeh style includes lens type, aperture size, etc. Only considering the blur amount variation, this task can be converted to bokeh rendering or defocus deblurring. If the aperture size of source lens is small while the counterpart of target lens is large, this task is similar to bokeh rendering, which artificially blurs an all-in-focus image. If transposing the source lens and target lens, this task then changes into defocus deblurring, which sharpens a blurred image. Since there are no proposed methods in terms of bokeh effect transformation, we introduce the recent work of bokeh rendering and defocus deblurring in this section.

2.1. Bokeh Rendering

From the perspective of the input form, we can categorize bokeh rendering task into automatic bokeh rendering and controllable bokeh rendering. The former one indicates that we can only input a single shallow depth-of-field image to perform rendering without providing other information. Thus, the method requires to automatically perceive the focused object of the input image and blurred background according to potential depth relationship. With the introduce of a large-scale bokeh dataset EBB! [8], many end-to-end networks [5, 8–11, 17, 20] have been proposed to specialize in the automatic bokeh rendering. Ignatov *et al*. [8] propose a PyNet-based framework which has an inverted pyramidal shape and processes images at seven different scales. Qian *et al*. [20] propose a two-stage Glass-Net, along with a GAN [7] loss to enhance the perceptual quality of the rendered results. Luo and Peng *et al*. [17] divide the task into three sub tasks: defocus hallucination, radiance virtualization and weighted layered rendering to increase the interpretability and performance in areas with large blur amount.

Controllable bokeh rendering, on the other hand, allows extra inputs, *e.g*., the measured or predicted disparity map and some controlling parameters, including blur amount parameter and refocused disparity. Nevertheless, the output is also required to be adjusted freely according to the different controlling parameters. The early methods typically adopt a strategy of layered rendering [3, 32], which decomposes the scene into multiple layers and blurs each of them independently before compositing them from back to front in order. However, these methods are prone to cause boundary artifacts. To tackle these problems, deep learning based methods have been proposed in recent years. Xiao *et al*. [30] introduce a new dataset synthesized by Unity and propose a framework to render images in low resolution. Wang *et al*. [26] propose a bokeh rendering system which consists of depth prediction, lens blur, and guided upsampling modules to process high-resolution images. Peng *et al*. [18] combine the advantages of classical rendering and neural rendering with a dual-stream framework. Peng *et al*. [19] further propose an MPI-based framework to specialize in rendering realistic partial occlusion effects.

2.2. Defocus Deblurring

Defocus deblurring, as a reverse task of bokeh rendering, is more challenging. Classical defocus deblurring methods typically first estimate a defocus map [12, 24, 33], followed by a non-blind deconvolution [6, 13, 15]. These methods perform poorly and produce ringing artifacts especially for complicated scenes due to the inaccurate estimated defocus map and naïve deconvolution operation.

Abuolaim and Brown [2] first propose a dual-pixel defocus deblurring method based on neural networks and introduce a corresponding dataset. For each scene, the dataset contains a pair of left/right dual-pixel (DP) images with large defocus blur and a single image with small defocus blur. They further improve the performance by adding DP data produced synthetically and extend the task to video application [1]. To address single defocus deblurring, Lee *et*

Figure 2. Images from the first row and the second row are captured by the lens with f-number of 16 and 1.8, respectively. Although the blur level of the right image group is larger than the left one, the blur ratio from f16.0 to f1.8 or from f1.8 to f16.0 is the same.

al. [14] propose a novel iterative filter adaptive network cooperated with an auxiliary reblurring module, which significantly boosts the deblurring quality. Son *et al*. [23] propose effective and lightweight kernel sharing parallel atrous convolution block to simulate spatially varying inverse kernels. Considering the spatial misalignment of the captured dataset, Ruan *et al*. [21] first pretrain the model on a light field dataset where the synthesized bokeh images are completely aligned with all-in-focus images, and then finetune the model on the captured dataset. On the other hand, Li *et al*. [16] adopt a deblurring module which incorporates a bidirectional optical flow-based deformation to constrain the spatial consistency. They also jointly train a spatially invariant reblurring module to further improve the performance.

3. Proposed Method

To tackle the bokeh effect transformation task, we propose SBTNet. In the following, we first explain our understanding of bokeh effect transformation and introduce a concept of "blur ratio" in Sec. 3.1. We then detail the structures of different modules in Sec. 3.2. Finally, we describe our multi-stage training and inference process in Sec. 3.3.

3.1. Bokeh Effect Transformation

Bokeh effect transformation mainly includes the transformation of lens type and blur amount. In general, the lens type is provided directly by the training sample, *e.g*., from Sony to Canon, while the blur amount of each image is hard to say since it is spatially variant and is related to multiple factors, including the aperture size, focal length, depth of focus, and the distance from camera to the scene. If encoding all of these factors and feeding them into the network, it will be hard for network to learn without a large amount of data. Fortunately, for the bokeh effect transformation task, the focused object is typically constant, enabling us to define a concept of blur ratio to simplify and represent the transformation of the blur amount.

Specifically, for a bokeh image, the blur amount of different areas can be represented by a defocus map S [25]:

$$S = \frac{l^2}{f}|D - d_f|\,, \tag{1}$$

where l is lens's focal length. f is lens's f-number, which is equal to the ratio of the focal length to the diameter of the entrance pupil. D is the disparity map or inverse depth map of the scene, and d_f is the disparity of focus. For a particular scene with an identical focused object but different captured lens, the blur ratio η of the two lenses can be defined by

$$\eta = \frac{S_t}{S_s} = \frac{l_t^2 f_s}{l_s^2 f_t}, \tag{2}$$

where subscript t and s denote the target lens and the source lens, respectively. Therefore, no matter what scene to shoot and how much distance between the lens and the captured scene, the blur ratio is determined as long as the focal length and f-number of the source lens and target lens are provided. Particularly, if $l_s = l_t$, as in the dataset BETD [4], η can be further simplified into $\frac{f_s}{f_t}$, as shown in Fig. 2.

3.2. Architecture of SBTNet

Alpha Extraction of Focused Object. The architecture of SBTNet is shown in Fig. 3. We first predict an alpha map of the focused object by AlphaNet, which facilitates preserving the sharp boundaries of focused objects in the transformed results. To extract more global information, we implement AlphaNet with a U-Net architecture where the bottom layers are replaced with consecutive CrosssFormer blocks [27] with long short distance attention.

Cat-Eye Effect Simulation. The cat-eye effect of camera lens represents that the bokeh balls are not circular at the corners of a captured image as shown in Fig. 4. To simulate

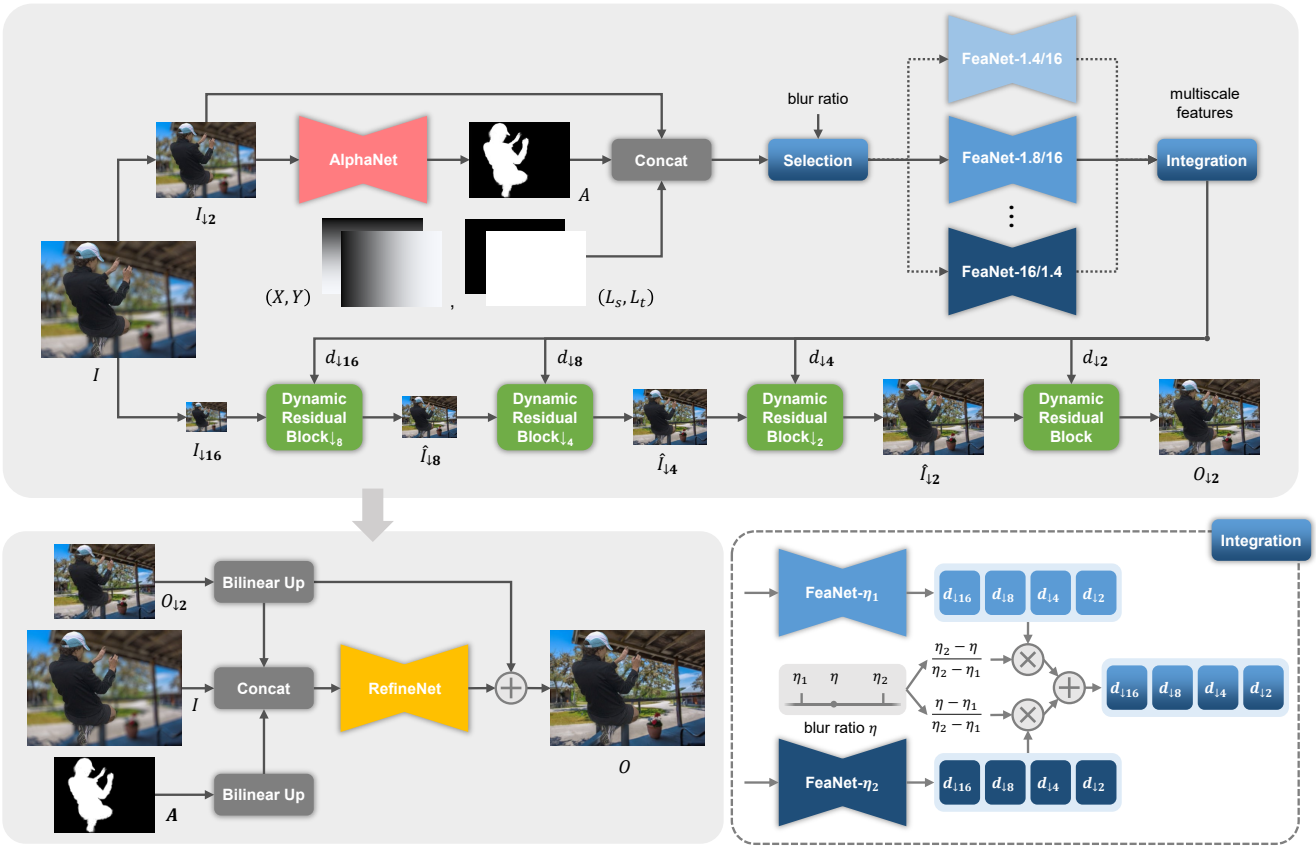


Figure 3. The architecture of SBTNet. AlphaNet first predicts an alpha map of the focused object. A coordinate map (X, Y) is considered to reflect the cat-eye effect of different positions. The lens type of the source image and the target image is encoded into a two-channel one-hot map (L_s, L_t) as an extra input to perform lens type transformation. Subsequently, according to different blur ratios, specific FeaNets are selected to extract multi-scale features, which are then integrated and fed into consecutive dynamic residual blocks to obtain a transformed image in half resolution. RefineNet is finally applied to obtain a full-resolution result.

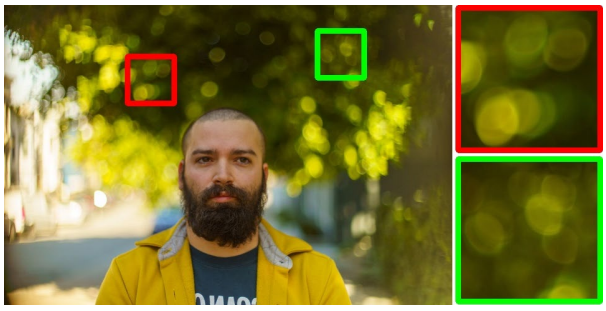


Figure 4. Cat-eye effect of the captured bokeh images [4]. The shape and direction of bokeh balls are not the same in different positions.

this effect, we add a two-channel coordinate map (X, Y) as an extra input, which reflects the degree of the cat-eye effect in different positions. We show in experiments that this coordinate map can enhance the performance both quantitatively and qualitatively.

Selective Bokeh Effect Transformation. To perform the transformation of lens type, we encode the lens type of the source image and the target image to a two-channel one-hot map (L_s, L_t) as an additional input of the subsequent network. Since there are only 2 lens types, *i.e.*, Sony and Canon in our training dataset, 2 channels are sufficient. Specifically, we set the value to 0 for Sony lens and 1 for Canon lens.

To learn the transformation of blur amount, we design a feature selection and integration strategy. Specifically, we first apply several parallel FeaNets with the same architecture to extract the multi-scale features corresponding to different blur ratios. For example, FeaNet-1.8/16 indicates that the lens's f-number of the source image is 16, while the counterpart of the target image is 1.8. In practice, during training, we select a particular FeaNet for each training sample, and during inference, we can integrate the features of two neighboring FeaNets, so that we can obtain the final result corresponding to an intermediate blur ratio. The pipeline of this integration process is shown in Fig. 3. Assume the blur ratio is η, and the blur ratios of the corre-

sponding neighboring FeaNets are η_1 and η_2, we can calculate the distributed weights for the multi-scale features of FeaNet-η_1 and FeaNet-η_2 by

$$\text{FeaNet-}\eta_1 : \frac{\eta_2 - \eta}{\eta_2 - \eta_1}, \quad \text{FeaNet-}\eta_2 : \frac{\eta - \eta_1}{\eta_2 - \eta_1}. \quad (3)$$

The actual relationship may not be linear, but this strategy indeed supports the blur amount transformation with the intermediate blur ratio. Subsequently, we use 4 dynamic residual modules to obtain results progressively. Through experiments, we verify that compared with integration in image level, integration in feature level performs better and is more efficient as we only need to run once the dynamic residual blocks. The architecture of FeaNet and dynamic residual module are borrowed from DRBNet [21], but we modify the basic channel from 32 to 64, which increases the capacity of the network.

Guided Upsampling and Refinement. The above predicted alpha map and the transformed image are both in half resolution, so we further design RefineNet to upsample and refine the transformed image. The input of RefineNet is the concatenation of the original input image and the bilinearly upsampled alpha map and the transformed image. The predicted residual is then added to the transformed image to obtain the final result. The architecture of RefineNet is similar to FeaNet, but the basic channel is set to 32.

3.3. Training and Inference Details

Multi-Stage Training. We implement our model with PyTorch. All of the experiments are conducted on 4 NVIDIA GeForce GTX 1080 Ti GPUs.

Our training contains 3 stages, where Adam optimizer is used for optimization. RefineNet is disabled during the first 2 stages. Compared with the bokeh transformation from small blur to large blur, the opposite process is much more difficult for network to learn. Thus, we only use the data where the f-number of source images is 1.8 and the f-number of target images is 16.0 at stage 1, which indicates that only FeaNet-1.8/16 is active. We augment the input image with random cropping and horizontal flipping. The learning rate is set to 10^{-4}. The model is trained for 300 epochs with a batch size of 8.

At stage 2, we initialize the parameters of all FeaNets with the parameters of FeaNet-1.8/16. All of the training data are used, and with the input of different f-number pairs, the corresponding FeaNet is active. The learning rate is set to 10^{-4} for all FeaNets and 10^{-5} for other structures. The model is trained for 150 epochs with a batch size of 64.

At stage 3, RefineNet is active and the parameters except for RefineNet are fixed. Empirically, we replace the alpha map feeding to RefineNet with an all-zero map with probability of 0.5 to increase the generalization ability. The learning rate is set to 10^{-4}. The model is trained for 100 epochs with a batch size of 8.

Table 1. Quantitative results of NTIRE 2023 Bokeh Effect Transformation Challenge [4]. We unofficially rank different teams by LPIPS. Note that "Base Results" and "EBokehNet [22]" are both proposed by the competition organizers. The best performance is in **boldface**.

Team	PSNR↑	SSIM↑	LPIPS↓
AIA-Smart (Ours)	34.572	0.9361	**0.0966**
Samsung Research China	**35.264**	**0.9362**	0.0985
NUS-LV-Bokeh	32.326	0.9333	0.1076
IPAL-Bokeh	32.076	0.9324	0.1076
BokehOrNot	32.288	0.9327	0.1130
BIGbaodan	30.327	0.9281	0.1249
IR-SDE	30.866	0.9297	0.1301
JiXiangNiu	27.970	0.9213	0.1542
Base Results	28.599	0.9128	0.1878
EBokehNet (Organizers)	34.543	0.9350	0.1039

Loss Functions. During the first 2 training stages, we use the loss function as follow:

$$\begin{aligned}\mathcal{L}_{1,2} = &\mathcal{L}_{\ell_1}(O_{\downarrow 2}, O^*_{\downarrow 2}) + \mathcal{L}_{\ell_1}(\nabla O_{\downarrow 2}, \nabla O^*_{\downarrow 2}) \\ &+ \mathcal{L}_{SSIM}(O_{\downarrow 2}, O^*_{\downarrow 2}) + \mathcal{L}_{BCE}(A_{\downarrow 2}, A^*_{\downarrow 2}), \end{aligned} \quad (4)$$

where ground-truth maps are marked with a superscript $*$. ∇ denotes the gradient domain. $O_{\downarrow 2}$ and $A_{\downarrow 2}$ are the transformed result and the predicted alpha map in half resolution. $\mathcal{L}_{\ell_1}$ is the ℓ_1 loss. $\mathcal{L}_{SSIM}$ is the structural similarity (SSIM) loss [28]. $\mathcal{L}_{BCE}$ is the binary cross entropy (BCE) loss.

At stage 3, only RefineNet is supervised in full resolution, so the loss function is set to

$$\begin{aligned}\mathcal{L}_3 = &\mathcal{L}_{\ell_1}(O, O^*) + \mathcal{L}_{\ell_1}(\nabla O, \nabla O^*) \\ &+ \mathcal{L}_{SSIM}(O, O^*). \end{aligned} \quad (5)$$

Inference. We observe that for real-world images, the predicted alpha map degrades significantly. Most blurred background areas are perceived as the focused foreground, and the bokeh style of these areas does not transform. The reason may be that the focused objects of the training dataset are pasted on the captured background manually, which has a large gap from real-world images. To increase the generalization ability, we disable AlphaNet and set the alpha map to an all-zero map for real-world images during inference.

Notably, we can even perform bokeh transformation with arbitrary blur ratio which does not exist in the training dataset via interpolating the features of two neighbouring FeaNets.

Canon50mmf1.4 → Sony50mmf16.0 Canon50mmf1.8 → Sony50mmf16.0 Canon50mmf1.4 → Sony50mmf1.8

Sony50mmf16.0 → Canon50mmf1.4 Sony50mmf16.0 → Canon50mmf1.8 Sony50mmf1.8 → Canon50mmf1.4

Figure 5. Qualitative results of synthesized images in test subset.

Canon50mmf1.4 → Sony50mmf1.8 Canon50mmf1.4 → Sony50mmf1.8 Canon50mmf1.4 → Sony50mmf1.8

Sony50mmf1.8 → Canon50mmf1.4 Sony50mmf1.8 → Canon50mmf1.4 Sony50mmf1.8 → Canon50mmf1.4

Figure 6. Qualitative results of real-world images in test subset.

4. Experiments

4.1. Dataset and Metrics

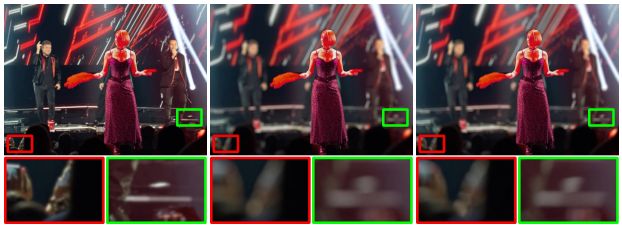
NTIRE 2023 Bokeh Effect Transformation Challenge [4] introduces a novel dataset BETD. The training subset contains 20k samples and each sample consists of a source image, a target image, an alpha map and related metadata. The source and target images are synthesized by a foreground RGBA image which serves as a focused object and a background image, artificially blurred with different lenses. The metadata lists the lens type, focal length, f-number of the source and target lenses, and an blur strength indicator for the scene. The focal length is always 50mm, and there exist 2 lens types, *i.e.*, Sony and Canon, and 7 types of f-number pairs, *i.e.*, $(1.4, 16.0)$, $(1.8, 16.0)$, $(1.4, 1.8)$, $(1.8, 1.8)$, $(1.8, 1.4)$, $(16, 1.8)$ and $(16, 1.4)$. The validation and test subsets respectively contain 500 and 180 samples without target images and alpha maps. Note that the test subset contains 95 synthesized images and 85 real-world images, where the latter ones are captured by the same lenses as for the former ones and have the same structure of sharp foregrounds in front of blurred backgrounds.

To select the best model during training and to facilitate subsequent experiments, we split an extra validation subset which contains 200 samples with ground truths from the original training subset. This validation subset is termed as Val200.

During evaluation, aside from the commonly used met-

Table 2. Ablation study on Val200 w/ and w/o the input of coordinate maps.

Coordinate Map	PSNR↑	SSIM↑	LPIPS↓
w/o	43.567	0.9892	0.0352
w/	**45.627**	**0.9946**	**0.0331**



Source w/o Coord. w/ Coord.

Figure 7. Visualized results w/ and w/o the input of coordinate maps. In this example, the blur ratio is larger than 1.

rics PSNR and SSIM, LPIPS [31] is also added to reflect the perceptual quality of the transformed results.

4.2. Quantitative and Qualitative Results

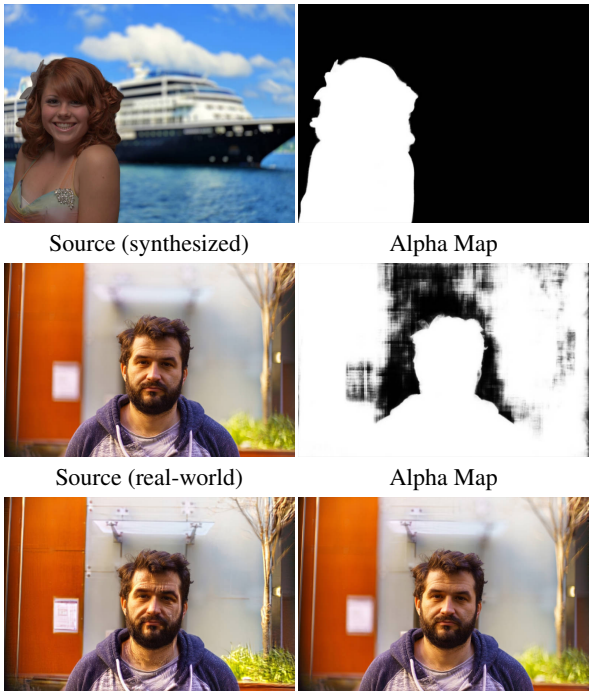
Our model is proposed to participate in the NTIRE 2023 Bokeh Effect Transformation Challenge [4]. Table 1 lists the quantitative metrics of the approaches proposed by different teams. One can observe that we rank first in LPIPS and rank second in PSNR and SSIM, demonstrating that our method can render more perceptually realistic transformed results compared with other methods. Since no codes of related approaches have been published so far, we only visualize some results of our method in Fig. 5 and Fig. 6. One can observe that the bokeh transformation effect is prominent, especially for the settings from a relatively sharp image to a blurred image as well as the opposite one.

4.3. Analyses of SBTNet

To demonstrate the necessity of individual modules and the generalization ability of SBTNet, we conduct comprehensive analyses on Val200.

Coordinate Map. Due to the cat-eye effect of captured bokeh images, the blurring patterns differ in different areas. As we randomly crop images during training, the network cannot perceive the position of the cropped patch without inputting a coordinate map. From Table 2, adding the coordinate map significantly enhances the performance of bokeh transformation. As shown in Fig. 7, if we render a blurred image from a sharp image, we can produce prominent cat-eye effect when adding the coordinate map.

AlphaNet. AlphaNet is designed to predict the focused object of the source image, so that it will not be changed in the transformed result. However, from Fig. 8, despite of



Source (synthesized) Alpha Map

Source (real-world) Alpha Map

w/o AlphaNet w/ AlphaNet

Figure 8. Visualized results of the synthesized image and the real-world image w/ and w/o using AlphaNet. In this example, the blur ratio is smaller than 1.

good performance on synthesized training data, the quality of the predicted alpha map is poor when dealing with real-world images. The reason may be the bokeh style gap on background between synthesized images and real-world images, so AlphaNet recognizes the blurred background into focused areas, leading to incorrectly unchanged bokeh effect in transformed results. Although replacing the predicted alpha map with all-zero map can alleviate this problem, it causes artifacts within the focused area. Therefore, our method still has room for improvement.

Feature Selection and Integration. During inference, we can obtain the transformed result of arbitrary blur ratio by the integration of two neighboring FeaNets. To verify that the integration in feature level is superior to the integration in image level, we compare the two integration strategies in Table 3. The transformed results from f1.8 to f16.0 can be computed by integrating the results from f1.4 to f16.0 and the results from f1.4 to f1.8. The transformed results from f16.0 to f1.8 can be computed in a similar way. The statistics are consistent with our expectations. Meanwhile, integration in feature level is more time-saving than in image level due to running once the dynamic residual blocks.

Model Complexity. In Table 4, we list the parameters of different modules in SBTNet. Since there are 7 types of f-number pairs in the dataset BETD [4], we set 7 FeaNets

Table 3. Comparison of integration in image level and feature level.

Integration Mode	PSNR↑	SSIM↑	LPIPS↓
Canon50mmf1.8 → Sony50mmf16.0			
Direct Image Level	35.755	0.9850	0.0684
Feature Level	**35.768**	**0.9851**	**0.0675**
Sony50mmf16.0 → Canon50mmf1.8			
Image Level	37.282	0.9894	0.0752
Feature Level	**38.005**	**0.9921**	**0.0555**

Table 4. Parameters of different modules in SBTNet.

Modules	AlphaNet	FeaNets	DRB	RefineNet	Total
Params (M)	44.4	27.8×7	18.9	7.2	265

for each of them. In practice, according to the calculated blur ratio, we use one or two FeaNets at a time. During evaluation, it takes about 1.3s to process a 1920×1440 image on a NVIDIA GeForce GTX 1080 Ti GPU.

5. Conclusion

In this paper, we present SBTNet for the bokeh effect transformation task. To simulate the cat-eye effect of real lens, we add a coordinate map as an extra input. Besides, since bokeh effect transformation includes the transformation of lens type and blur amount, we additionally input a two-channel lens type map and design a feature selection and integration strategy to handle different blur ratios. Our solution gets the best LPIPS metric in NTIRE 2023 Bokeh Effect Transformation Challenge, demonstrating the effectiveness and high perceptual quality of our method. Despite of this, further study is still required to better perceive the spatially variant blur amount of real-world images and to produce more accurate and smooth transformed results with arbitrary blur ratios.

References

[1] Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Improving single-image defocus deblurring: How dual-pixel images help through multi-task learning. In *Proc. IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1231–1239, 2022. 2

[2] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *Proc. European Conference on Computer Vision (ECCV)*, pages 111–126. Springer, 2020. 2

[3] Benjamin Busam, Matthieu Hog, Steven McDonagh, and Gregory Slabaugh. Sterefo: Efficient image refocusing with stereo vision. In *Proc. IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 0–0, 2019. 1, 2

[4] Marcos V Conde, Manuel Kolmet, Tim Seizinger, Tom E Bishop, and Radu Timofte. Lens-to-lens bokeh effect transformation. ntire 2023 challenge report. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. 1, 2, 3, 4, 5, 6, 7

[5] Saikat Dutta, Sourya Dipta Das, Nisarg A Shah, and Anil Kumar Tiwari. Stacked deep multi-scale hierarchical network for fast bokeh effect rendering from a single image. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2398–2407, 2021. 1, 2

[6] DA Fish, AM Brinicombe, ER Pike, and JG Walker. Blind deconvolution by means of the richardson–lucy algorithm. *JOSA A*, 12(1):58–65, 1995. 2

[7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 2

[8] Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 418–419, 2020. 1, 2

[9] Andrey Ignatov, Jagruti Patel, Radu Timofte, Bolun Zheng, Xin Ye, Li Huang, Xiang Tian, Saikat Dutta, Kuldeep Purohit, Praveen Kandula, et al. Aim 2019 challenge on bokeh effect synthesis: Methods and results. In *Proc. IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3591–3598. IEEE, 2019. 1, 2

[10] Andrey Ignatov, Radu Timofte, Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, Jian Cheng, Juewen Peng, et al. Aim 2020 challenge on rendering realistic bokeh. In *Proc. European Conference on Computer Vision Workshops (ECCVW)*, pages 213–228. Springer, 2020. 1, 2

[11] Andrey Ignatov, Radu Timofte, Jin Zhang, Feng Zhang, Gaocheng Yu, Zhe Ma, Hongbin Wang, Minsu Kwon, Haotian Qian, Wentao Tong, et al. Realistic bokeh effect rendering on mobile gpus, mobile ai & aim 2022 challenge: Report. In *Proc. European Conference on Computer Vision Workshops (ECCVW)*, pages 153–173. Springer, 2023. 1, 2

[12] Ali Karaali and Claudio Rosito Jung. Edge-based defocus blur estimation with adaptive scale selection. *IEEE Transactions on Image Processing (TIP)*, 27(3):1126–1137, 2017. 2

[13] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. *Advances in neural information processing systems*, 22, 2009. 2

[14] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2034–2042, 2021. 3

[15] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70–es, 2007. 2

[16] Yu Li, Dongwei Ren, Xinya Shu, and Wangmeng Zuo. Learning single image defocus deblurring with misaligned training pairs. *arXiv preprint arXiv:2211.14502*, 2022. 3

[17] Xianrui Luo, Juewen Peng, Ke Xian, Zijin Wu, and Zhiguo Cao. Bokeh rendering from defocus estimation. In *Proc. European Conference on Computer Vision Workshops (ECCVW)*, pages 245–261. Springer, 2020. 1, 2

[18] Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. Bokehme: When neural rendering meets classical rendering. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16283–16292, 2022. 1, 2

[19] 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 *Proc. European Conference on Computer Vision (ECCV)*, pages 590–607. Springer, 2022. 1, 2

[20] Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, and Jian Cheng. Bggan: Bokeh-glass generative adversarial network for rendering realistic bokeh. In *Proc. European Conference on Computer Vision (ECCV)*, pages 229–244. Springer, 2020. 1, 2

[21] Lingyan Ruan, Bin Chen, Jizhou Li, and Miuling Lam. Learning to deblur using light field generated and real defocus images. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16304–16313, 2022. 3, 5

[22] Tim Seizinger, Marcos V Conde, Manuel Kolmet, Tom E Bishop, and Radu Timofte. Efficient multi-lens bokeh effect rendering and transformation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2023. 5

[23] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2642–2650, 2021. 3

[24] Yu-Wing Tai and Michael S Brown. Single image defocus map estimation using local contrast prior. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1797–1800. IEEE, 2009. 2

[25] Neal Wadhwa, Rahul Garg, David E Jacobs, Bryan E Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018. 1, 3

[26] 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. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018. 2

[27] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *International Conference on Learning Representations (ICLR)*, 2022. 3

[28] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 13(4):600–612, 2004. 5

[29] Ke Xian, Juewen Peng, Chao Zhang, Hao Lu, and Zhiguo Cao. Ranking-based salient object detection and depth prediction for shallow depth-of-field. *Sensors*, 21(5):1815, 2021. 1

[30] Lei Xiao, Anton Kaplanyan, Alexander Fix, Matthew Chapman, and Douglas Lanman. Deepfocus: Learned image synthesis for computational displays. *ACM Transactions on Graphics (TOG)*, 37(6):1–13, 2018. 2

[31] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. 7

[32] Xuaner Zhang, Kevin Matzen, Vivien Nguyen, Dillon Yao, You Zhang, and Ren Ng. Synthetic defocus and look-ahead autofocus for casual videography. *ACM Transactions on Graphics (TOG)*, 38:1 – 16, 2019. 1, 2

[33] Shaojie Zhuo and Terence Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011. 2