


An α -Matte Boundary Defocus Model-Based Cascaded Network for Multi-focus Image Fusion

Haoyu Ma , Qingmin Liao, *Senior Member, IEEE*, Juncheng Zhang, Shaojun Liu, Jing-Hao Xue, *Member, IEEE*

Abstract—Capturing an all-in-focus image with a single camera is difficult since the depth of field of the camera is usually limited. An alternative method to obtain the all-in-focus image is to fuse several images that are focused at different depths. However, existing multi-focus image fusion methods cannot obtain clear results for areas near the focused/defocused boundary (FDB). In this paper, a novel α -matte boundary defocus model is proposed to generate realistic training data with the defocus spread effect precisely modeled, especially for areas near the FDB. Based on this α -matte defocus model and the generated data, a cascaded boundary-aware convolutional network termed MMF-Net is proposed and trained, aiming to achieve clearer fusion results around the FDB. Specifically, the MMF-Net consists of two cascaded subnets for initial fusion and boundary fusion. These two subnets are designed to first obtain a guidance map of FDB and then refine the fusion near the FDB. Experiments demonstrate that with the help of the new α -matte boundary defocus model, the proposed MMF-Net outperforms the state-of-the-art methods both qualitatively and quantitatively.

Index Terms—Image fusion, multi-focus, CNNs, defocus model.

I. INTRODUCTION

WHEN photos are taken with cameras, all-in-focus images are often desired as the output, in particular for a large number of computer vision tasks, such as localization, detection and segmentation [2]. However, it is usually hard to obtain an all-in-focus image from a single camera since the depth of field of the camera is limited [3]. Multi-focus image fusion is the approach to generate an all-in-focus image from several images taken of the same scene but focused at different depths, as shown in Fig. 1 via an example of the fused image obtained from two source images.

Existing multi-focus image fusion (MFIF) methods can be broadly categorized into three groups, i.e., transform domain algorithms, spatial domain algorithms, and convolutional neural network (CNN)-based algorithms [4].

Manuscript received October 29, 2019; revised June 26, 2020; accepted August 1, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61771276 and in part by the National Key Research and Development Program of China under Grant 2016YFB0101001. This work used a part of the Adobe Deep Matting dataset [1] for model training. The authors would like to thank the three reviewers for their helpful comments that improved this manuscript, particularly at this special time. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianbing Shen. (*Corresponding author: Shaojun Liu.*)

H. Ma, Q. Liao and J. Zhang are with Tsinghua Shenzhen International Graduate School and the Department of Electronic Engineering, Tsinghua University, Shenzhen 518055, China.

S. Liu is with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong, China (Email: liusj14@tsinghua.org.cn).

J.-H. Xue is with the Department of Statistical Science, University College London, WC1E 6BT, UK.



Fig. 1. An example of multi-focus image fusion with two source images; the fused image was obtained by our proposed MMF-Net.

The transform domain MFIF algorithms first decompose the source images and then fuse the results according to some handcrafted features. Typical transform domain MFIF algorithms include the nonsampled contourlet transform (NSCT) method [5], the sparse representation (SR) method [6], [7] and the combined NSCT-SR method [8]. Due to the imperfection of transformations and handcrafted features, these algorithms often produce nonrealistic results, even in the areas far away from the focused/defocused boundary (FDB).

The spatial domain MFIF algorithms include region-based methods and pixel-based methods. Region-based MFIF algorithms suffer from the blocking effect [9], [10]. Pixel-based MFIF algorithms first obtain a 0/1 discrete decision map for fusion [11] and then fuse the source images. The guided filtering (GF) method [12] and the dense SIFT (DSIFT) method [11] are typical pixel-based algorithms. Compared with the transform domain algorithms, the fusion results from the pixel-based algorithms are usually better. However, due to the defocus effect, none of the source images are clear in the areas near the FDB, and consequently, the fusion results of these methods are often unclear in these areas.

The CNN was first explored to extract defocus descriptors in a data-driven way in [13]. Existing neural network-based MFIF algorithms can be divided into two groups: decision map-based algorithms and end-to-end algorithms. Decision map-based algorithms [14]–[16] produce the decision map first as done in the pixel-based algorithms; consequently, similar to the pixel-based algorithms, they lead to the unclear FDB. End-to-end algorithms [17]–[19] directly obtain the fusion results, but the results are unfortunately not realistic, as in the transfer domain algorithms. Moreover, since it is hard to acquire a large number of all-in-focus ground truth images for training, data generation methods need to be adopted in these CNN-based algorithms [14]–[19]; however, these methods do not imitate the complex situation of the defocus spread near the FDB [8],

and thus some unnatural and unrealistic training data limit the performance of these networks.

In this paper, to address the issue of unsatisfactory fusion results near the FDB, we first present a discussion regarding the difficulties around the FDB. Then, an α -matte boundary defocus model is proposed to simulate the defocus spread effect near the FDB. Based on the α -matte defocus model and the generated training data, we develop a cascaded network for MFIF, which is called the Matte Model Fusion Net (MMF-Net). The technical underpinnings and contributions of our work are two-fold:

First, a novel α -matte boundary defocus spread model is proposed. Compared with existing defocus models, the proposed α -matte model is the first one to specifically model the difference in the defocus spread when the defocus occurs in the foreground or the background. Therefore, the α -matte model can generate simulated defocus images with the valid defocus spread near the FDB, which can be then used as training data to train deep neural networks.

Second, based on the proposed α -matte defocus model and the generated training data, a cascaded boundary-aware convolutional network termed MMF-Net is designed and trained to obtain clear fusion results in areas both far away from and near the FDB. Compared with the existing end-to-end CNN algorithms, the proposed MMF-Net generates a guidance map first, acquiring clear and realistic fusion results in areas far away from the FDB. Compared with the decision map-based CNN algorithms, the MMF-Net generates the areas near the FDB directly from the source images, similar to the end-to-end CNN algorithms, and thus achieves more reasonable and clearer fusion results than that achieved by any source images near the FDB.

Experiments demonstrate that on the benchmark dataset, the proposed MMF-Net outperforms the state-of-the-art methods, both qualitatively and quantitatively.

The rest of this paper is organized as follows. In section II, we present the issue to be addressed in this paper, introduce the proposed α -matte boundary defocus model, and conduct comparisons with the existing defocus models on a simulated scene. In section III, we present the proposed MMF-Net and discuss the corresponding loss functions. Experimental studies, including the method comparisons, are conducted in section IV, and conclusions and future work are drawn in section V.

II. α -MATTE BOUNDARY DEFOCUS MODEL

In this section, we first discuss the defocus spread effect around the focused/defocused boundary (FDB), and why it is difficult to contend with. Then, we briefly introduce and analyze two existing defocus models: the one-parameter defocus model [20] and our previous two-parameter defocus model [8]. Finally, a novel α -matte boundary defocus spread model is proposed based on the above discussion and analysis. Simulation experiments on a well-designed scene show that the proposed α -matte model can generate a defocus effect near the FDB much more realistically than the existing defocus models.



(a) Foreground focus

(b) Background focus

Fig. 2. Different defocus spread effects when the foreground or the background is out of focus, shown in enlarged real-world images. The in-focus boundary of the foreground object is labeled red.

A. The Focused/Defocused Boundary (FDB)

Existing CNN methods cannot obtain realistic and clear fusion results, particularly for the areas near the FDB. There are three main reasons for this issue.

First, the situations are quite different between patches far away from and near the FDB, and it is unwise to address an area near the FDB and an area far away from the FDB together, as stated in our previous work [21]. For the patches far away from the FDB, the patches are totally focused or defocused. Consequently, the defocus of the patch is homogeneous. In contrast, for patches near the FDB, both the focused area and defocused area exist. Therefore, it is hard to separate the focused area and the defocused area at the pixel level.

Second, there is a blurry area along the FDB, which is unclear in both source images A and B, because the defocus effect will spread out [8]. When the foreground object is in focus, but the background is defocused, the foreground object will not be influenced by the defocus of the background. As shown in Fig. 2(a), the foreground and the background are divided clearly by the yellow line. In contrast, when the foreground object is defocused, the defocus spread effect will lead to a blurry object larger than the original focused object, as shown in Fig. 2(b), in which we highlight the boundary of foreground objects in yellow to show the difference in the defocus spread. In the area between the two red lines, the defocus can be seen outside the yellow line, and the area inside the yellow line is influenced by the background as well. As a result, there is an area that is blurry in both source images A and B along the outside of the foreground objects; one is due to the defocus of the background, and the other is due to the defocus spread from the defocused foreground objects.

Third, when the foreground object is out of focus (as shown in Fig. 2(b)), it will be influenced slightly by the background on the inside of the original boundary as well, compared with Fig. 2(a). That is because the moving of the camera lens when changing the focal length will lead to a small change of the scene. As a result, although the scene varies very slightly, there will be a mismatch for areas around the FDB when a fusion is conducted on the source images.

The defocus spread effect of foreground objects makes it hard to obtain a clear result near the FDB. Some of the existing

methods [11], [14], [15] choose the pixel directly from one of the source images, and thus the fusion results near the FDB will be blurry and have artifacts. Some post processing methods [15], such as the guided filter [22], are implemented to derive smooth FDB but still remain unclear. Even using a weighted average of the source images as the fusion result [8], [12], [19], the blur will still remain. To address this issue, we need to carefully model the defocus spread for the areas near the FDB to generate a large number of realistic training images to train the neural networks for the MFIF. The performance of these data-driven methods is highly dependent on the training datasets; therefore, determining how closely the model can simulate the reality of this scenario is of vital importance.

There are several defocus spread models based on which the training datasets for CNN-based MFIF methods are generated. Existing models include the one-parameter defocus model [20] and the two-parameter defocus model [8]. Usually, these models are employed to generate a training dataset for MFIF methods. However, these models are not always valid, especially for the areas near the FDB.

B. One-parameter Defocus Model

The typical one-parameter 2D linear space-invariant defocus model [20] can be characterized with a space-invariant point spread function (PSF) [23]:

$$I(x, y) = h(x, y) \otimes f(x, y) + n(x, y), \quad (1)$$

where $I(x, y)$ is the defocused image at pixel (x, y) , $f(x, y)$ is the original image without the defocus effect, $n(x, y)$ is the additive noise, $h(x, y)$ is the defocus kernel, and \otimes denotes the convolution operator. In practice, the defocus kernel $h(x, y)$ is usually approximated with a 2D isotropic Gaussian kernel:

$$h(x, y) = G(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (2)$$

where σ is the standard deviation, which describes the defocus amount and is related to the distance between the object and the camera.

Several CNN-based MFIF methods such as [14], [15] employed this model to generate training data. In [15], the original images from the ImageNet dataset [24] are directly reblurred with a random Gaussian kernel for the defocus effect. This is a simple way to generate data, but the relation between the defocus amount and the depth is not considered. [14] noticed that usually the defocus did not occur over the whole image, and thus they reblurred the input image only in a predetermined area. However, the boundary of the predetermined area usually did not coincide with the real boundary of the object in the image.

C. Two-parameter Defocus Model

Since the defocus level is related to the depth between the objects and the camera, different objects can have different defocus levels. In our previous work on defocus map estimation [8], a two-parameter defocus model was proposed to describe the defocus spread effect for the area near the object boundaries, using a PSF with two parameters to describe the

different defocus levels on the two sides of a boundary. For an ideal 2D boundary,

$$\begin{aligned} f(x, y) &= f_A(x, y)u(ax + by + c) + \\ &f_B(x, y)u(-ax - by - c), \end{aligned} \quad (3)$$

and the defocused boundary will be

$$\begin{aligned} I(x, y) &= f_A(x, y)u(ax + by + c) \otimes h_A(x, y) + \\ &f_B(x, y)u(-ax - by - c) \otimes h_B(x, y), \end{aligned} \quad (4)$$

where $u(\cdot)$ is the step function, $ax + by + c = 0$ is the line corresponding to the boundary, and $f_A(x, y)$ and $f_B(x, y)$ are the original image areas at the different sides of the boundary, respectively. In [8], the defocus kernels $h_A(x, y)$ and $h_B(x, y)$ are approximated with two different 2D isotropic Gaussian kernels:

$$h_A(x, y) = G_A(x, y; \sigma_A) = \frac{1}{2\pi\sigma_A^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_A^2}\right), \quad (5)$$

$$h_B(x, y) = G_B(x, y; \sigma_B) = \frac{1}{2\pi\sigma_B^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_B^2}\right). \quad (6)$$

Taking advantage of this model, we generate the training data for MFIF in our previous work [21], in which the foreground objects or the background are first blurred with the Gaussian function and then spliced together. Consequently, the defocus level only changes alongside object boundaries, which is closer to reality than [14], [15]. However, this two-parameter model cannot describe the different FDB between an out-of-focus foreground (with an in-focus background) and an out-of-focus background (with an in-focus foreground).

D. An α -Matte Boundary Defocus Spread Model

As we have discussed above, existing defocus models focus on the intensity at each pixel rather than the defocus spread along the boundary of the foreground objects. To simulate the defocus effect, a new model should focus on several specific issues: the defocus spread across the FDB, the blurry area along the FDB, and the different spread situations of the FDB when defocus occurs in the foreground or the background. To address these issues, we first propose a novel α -matte boundary defocus spread model to simulate the defocus process. The proposed defocus model not only can be used to explain why the previous MFIF methods fail in the fusion of area near FDB, but also can be used readily to generate training data for MFIF, and the generated data will be not only similar to real scenes but also easy to be obtained than manual annotation.

In the proposed α -matte model, we assume that there is a transmission matte α_n for every surface S_n parallel to the focal plane, where $n(= 1, \dots, N)$ is the order of the surface. First, we assume that when a surface is in focus, for the surface without an object on it, the matte value is zero, and for the surface with objects that no light can pass through, the matte on the object pixel is one. Second, the defocus kernel $h_n(x, y)$ for a defocused surface $S_n(x, y)$ is a 2D isotropic Gaussian kernel $G(x, y; \sigma_n)$:

$$h_n(x, y) = G(x, y; \sigma_n) = \frac{1}{2\pi\sigma_n^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_n^2}\right). \quad (7)$$

Third, the defocus effects are the same for the RGB surface S_n and the matte α_n :

$$\begin{aligned}\alpha_n^0(x, y) &= h_n(x, y) \otimes \alpha_n^c(x, y) \\ &= G(x, y; \sigma_n) \otimes \alpha_n^c(x, y),\end{aligned}\quad (8)$$

$$\begin{aligned}S_n(x, y) &= h_n(x, y) \otimes S_n^c(x, y) \\ &= G(x, y; \sigma_n) \otimes S_n^c(x, y),\end{aligned}\quad (9)$$

where α_n^c is the clear matte on the in-focus clear surface S_n^c , and α_n^0 denotes the matte before considering any defocus spread effects from the objects in front of S_n .

As have been shown in Fig. 2, the out-of-focus objects in the front can affect objects in the back, but the out-of-focus objects in the back cannot affect objects in the front. Therefore, we make the clear RGB surface S_n^c and the clear matte α_n^c out of focus one by one from near to far. The final matte α_n will be an aggregation of all effects from those mattes in the front, that is, the intersection between its own α_n^0 and the complementary set for the summation of all the mattes in the front:

$$\alpha_n = \alpha_n^0 \left(1 - \sum_{t=0}^{n-1} \alpha_t\right), \quad \text{with } \alpha_0 = 0, \quad n = 1, \dots, N. \quad (10)$$

This means that the defocus of the matte would only influence the mattes behind it, as occurs in reality. Therefore, the captured image I at each pixel in the photo will be a summation of the pixel values on all surfaces S_n :

$$I = \sum_{n=1}^N I_n = \sum_{n=1}^N \left[\left(1 - \sum_{t=0}^{n-1} \alpha_t\right) S_n \right]. \quad (11)$$

Particularly, image I with only two valid surfaces (foreground surface S_{FG} and background surface S_{BG}) is

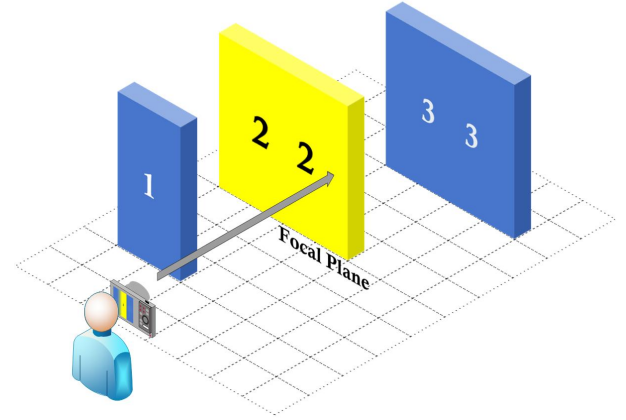
$$I = S_{FG} + (1 - \alpha_{FG})S_{BG}, \quad (12)$$

where α_{FG} is the matte of the foreground, and this is the model we used for data generation.

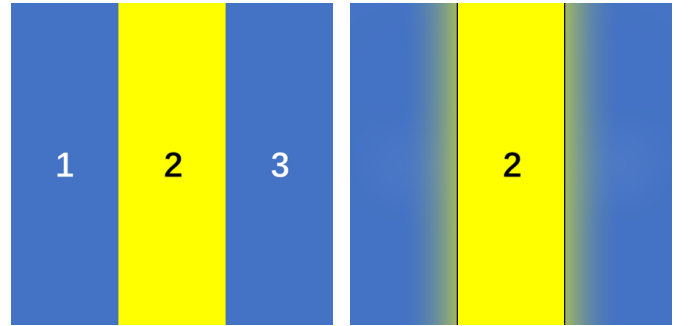
E. Comparison of Defocus Models

To show the difference between the proposed α -matte defocus spread model and the existing defocus models, we simulate it with a simple scene. In the simulated scene, there are three objects, as shown in Fig. 3(a). Object 1 is close to the camera, object 3 is far away from the camera, and object 2 is set between object 1 and object 3. The all-in-focus image is as shown Fig. 3(b), which cannot be taken by the camera with limited focal length directly. We focus the camera on the object 2, so the defocus spread effect near the FDB varies. For the boundary between object 1 and object 2, the foreground is out of focus and the background is in focus. In contrast, for the boundary between object 2 and object 3, the foreground is in focus and the background is out of focus.

According to the one-parameter defocus model [20], the simulated image will appear as in Fig. 3(c). The model can simulate the defocus effect in object 1 and object 3, which are out of focus. However, the one-parameter defocus model cannot show the defocus spread effect at all: in the area near the FDB boundary in object 2, there is no influence of

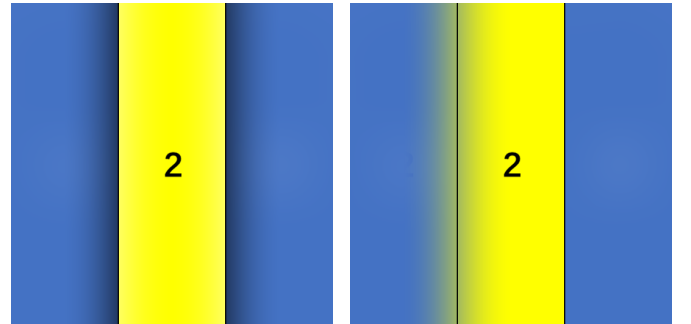


(a) The simulated scene



(b) All-in-focus image

(c) One-parameter defocus model [20]



(d) Two-parameter defocus model [8]

(e) α -matte defocus model

Fig. 3. Comparison of the defocus models in the simulated scene with the camera focusing on object 2. The boundary between object 1 and object 2 is different from that between object 2 and object 3. The proposed α -matte model precisely simulates the defocus spread effect across the boundary between object 1 and object 2 (foreground out of focus), as well as the clear boundary between object 2 and object 3 (foreground in focus). The black lines are added to show the original boundary of clear objects.

defocus at all. Moreover, this model does not reflect whether the defocus object is in the front of or behind an in-focus object, with the FDB all the same, as shown in Fig. 3(c).

According to the two-parameter defocus model [8], the simulated image is as shown in Fig. 3(d). The defocus spread effect can be simulated on both sides of the FDB, but this defocus model suffers from the anti-gradient effect. Especially for the situation shown in this simulated scene, object 2 is in focus and shows no influence on object 1; and the defocus spread of object 1 is unclear since object 2 is in focus.

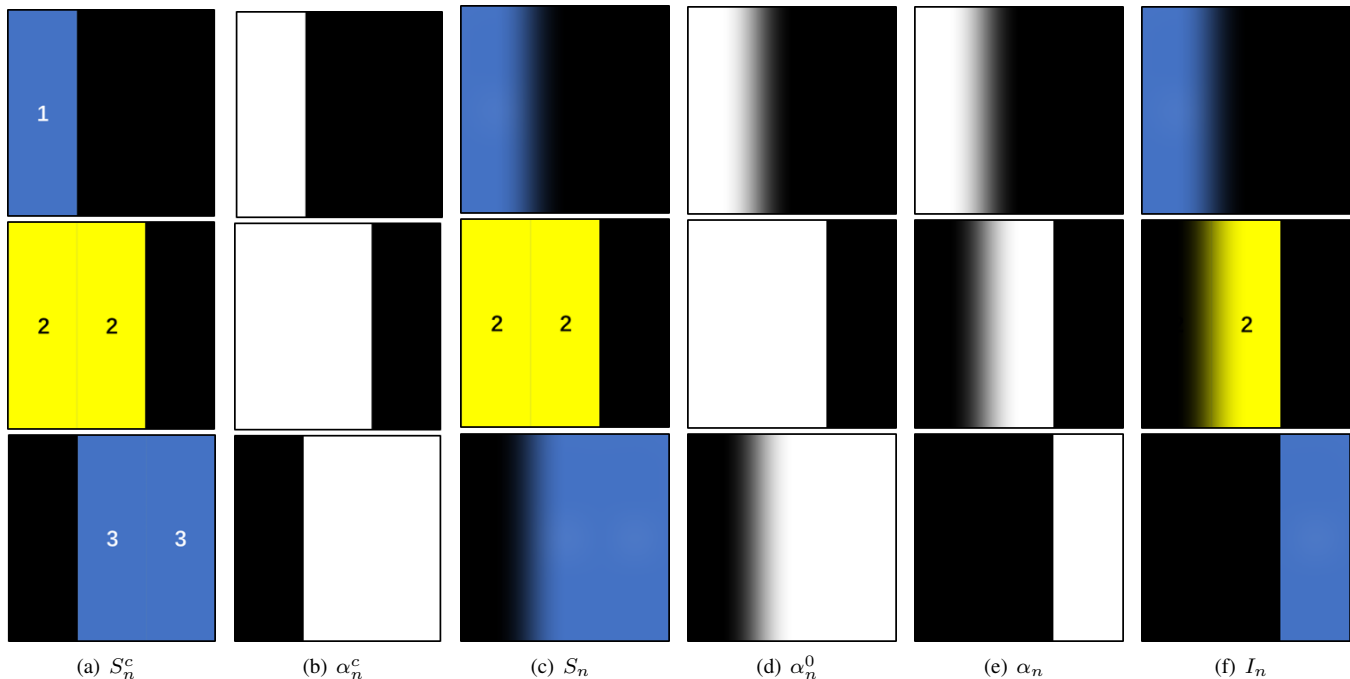


Fig. 4. Imaging process of the proposed α -matte defocus model in a stimulated scene with three surfaces from the top down. The captured image I (Fig. 3(e)) is the summation of the last column I_n .

Moreover, the two-parameter defocus model also does not reflect whether the defocus object is in the front of or behind an in-focus object, with the FDB all the same, as shown in Fig. 3(d).

The defocus simulated image of the proposed α -matte defocus spread model is shown in Fig. 3(e), from which we can observe two patterns.

First, for the FDB between object 1 and object 2, the defocus spread effect is on both sides: the effect in object 1 is due to the defocused object 1 and the moving camera lens, and the effect in object 2 is due to the spread effect of the defocused object 1. That is, because object 1 (foreground object) is out of focus and object 2 (background object) is in focus, the defocus will spread to the area of object 2 near the FDB, and the yellow color of object 2 will also have a slight influence on object 1. In other words, if object 2 has texture, we will notice that the defocus spread to object 1 is in fact the texture behind object 1 compared with the all-in-focus image, and the proposed α -matte model can also simulate the scene change when the focal length changes.

Second, for the FDB between the in-focus object 2 and the defocused object 3, the defocus spread has no effect on either side. Object 2 (foreground object) is in focus and object 3 (background object) is out of focus. Consequently, the defocus will not spread to the area of object 3 near the FDB since object 2 is in focus, and object 3 cannot influence object 2 as well, given that object 3, although out of focus, is behind the in-focus object 2.

In short, the proposed defocus model simulates very well the difference of the defocus spread when the defocus occurs in the foreground or the background.

Using the same simulated scene, we also show in Fig. 4 the

image composition process with the proposed α -matte defocus spread model. For every surface S_n parallel to the focal plane, the before-defocus object surfaces S_n^c is shown in Fig. 4(a), and the before-defocus matte α_n^c is shown in Fig. 4(b).

Then, the camera is set to focus on object 2, and object 1 and object 3 are out of focus. As we have mentioned above, a 2D isotropic Gaussian kernel $G(x, y; \sigma_n)$ is applied to the clear surface S_n^c and the clear matte α_n^c at the same time. The defocus exists when $n = 1$ or $n = 3$, and object 2 is in focus. The surfaces are shown in Fig. 4(c), and the mattes are shown in Fig. 4(d).

To compose the final image taken by the simulated camera, we first obtain the α_n (4(e)), which is the intersection between α_n^0 and the complementary set for the summation of all the mattes in the front. The calculation is done in a one-by-one manner from the surface in the front. After that, the images are generated using Equation (Fig. 11). The compositions of each surface are shown in Fig. 4(f). The final defocus image is the summation of I_1 , I_2 and I_3 , as we have shown in Fig. 3(e).

III. MMF-NET: THE MATTE MODEL BASED CASCADED FUSION NET

Based on the proposed defocus model and its generated training data, the cascaded convolutional fusion net (MMF-Net) is developed and trained. In this section, the structure of the proposed MMF-Net is first introduced, and then the loss functions used to train the MMF-Net are discussed. Our MMF-Net aims to achieve realistic and clear results in both the areas that are near and far away from the FDB.

A. The Network Structure

The structure of the proposed MMF-Net is shown in Fig. 5.

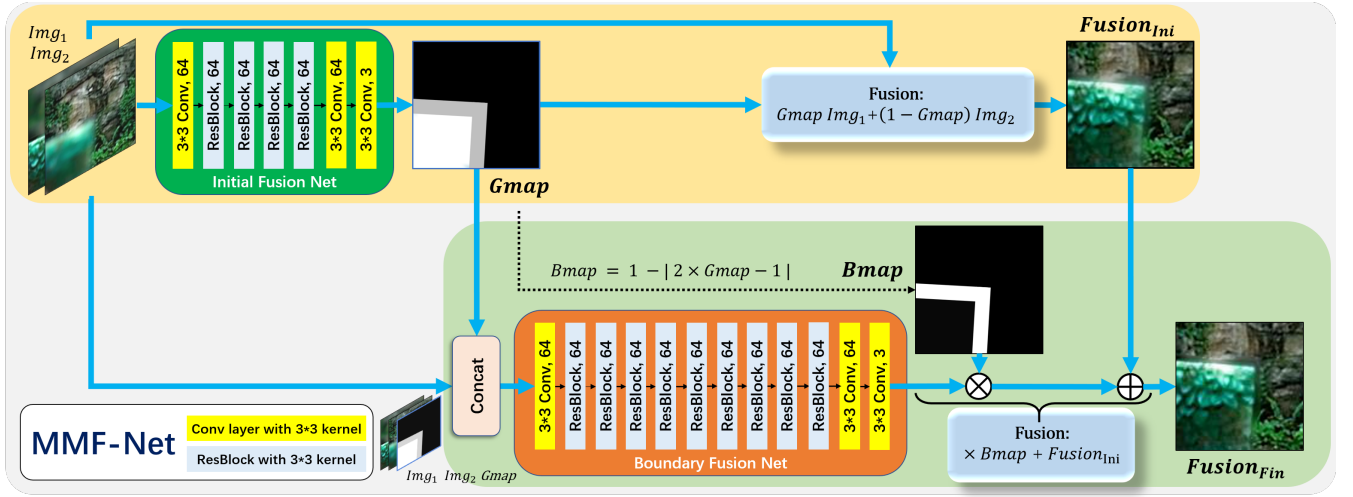


Fig. 5. The block diagram of the proposed cascaded boundary-aware multi-focus fusion network MMF-Net. The initial fusion subnet is implemented to derive the guidance map $Gmap$, and then the boundary fusion subnet is employed to help refine the fusion result near the FDB.

Two source images are first input into an initial fusion subnet aiming to generate a guidance map $Gmap$. In the guidance map, the pixel-wise value would be 1 if source image A is focused and source image B is defocused, whereas it would be 0 if source image A is defocused and source image B is focused, and the area near the FDB is given an α values of 0.5. Then, we use this guidance map to generate an initial fusion result:

$$Fusion_{Ini} = Gmap \times I_1 + (1 - Gmap) \times I_2, \quad (13)$$

where I_1 and I_2 are the two source images.

Then, the two source images are concatenated with the guidance map as the input of the boundary fusion subnet. The output of the boundary fusion subnet is masked and weighted by the boundary map $Bmap$ and then added to the initial fusion result. The boundary map is calculated with the guidance map as

$$Bmap = 1 - |2 \times Gmap - 1|. \quad (14)$$

In this way, regardless of which source image is focused, the value will be 0 for the areas far away from the FDB. Hence, only the pixels of boundary areas in the initial fusion result will be revised by the output of the boundary fusion subnet. That is, for the areas far away from the FDB, the final fusion results $Fusion_{Fin}$ of MMF-Net will be completely decided by the focused part of the source images. In the meantime, for the areas near the FDB, the final fusion results $Fusion_{Fin}$ are obtained through enhancing the initial fusion results by the output of the boundary fusion subnet, which is a hard task for existing methods as they do not specifically treat the FDB.

In our implementation, typical residual blocks [25] are employed. The initial fusion subnet contains 1 convolutional layer, 4 residual blocks, and 2 convolutional layers; the boundary fusion subnet contains 1 convolutional layer, 12 residual blocks, and 2 convolutional layers. The kernel size of every residual block is 64.

B. Loss Functions

The loss function that we use in the training process contains three components: the matte loss $Loss_{matte}$, the initial fusion loss $Loss_{Ini}$, and the weighted final fusion loss $Loss_W$:

$$Loss = \lambda_1 \times Loss_{matte} + \lambda_2 \times Loss_{Ini} + Loss_W, \quad (15)$$

where λ_1 and λ_2 are trade-off parameters.

First, for the initial fusion subnet, matte loss $Loss_{matte}$ and initial fusion loss $Loss_{Ini}$ are used:

$$Loss_{matte} = \text{mean}(|matte_{Ini} - matte_{GT}|), \quad (16)$$

$$Loss_{Ini} = \text{mean}((Fusion_{Ini} - Fusion_{GT})^2). \quad (17)$$

We choose the L1 norm for matte as it is a discrete value in the ground truth and choose the L2 norm for the fusion results as usual.

Second, in order to supervise the final fusion result more precisely, we use a weighted fusion loss $Loss_W$. The area near the FDB is much more difficult to be fused than the other areas, so its weight W should be larger than those for the areas far away from the FDB:

$$Loss_W = W \times \text{mean}((Fusion_{Fin} - Fusion_{GT})^2), \quad (18)$$

where the weight W is simply calculated as follows:

$$W = \frac{1 + (k - 1) \times (1 - |2 \times matte_{Ini} - 1|)}{k}, \quad (19)$$

where k is the weight contrast parameter of the FDB area. The max weight for the area near the FDB will be close to 1, as the matte values will be close to 0.5; and for the area far away from the FDB, the weight will be close to $1/k$, as the matte values will be close to either 1 or 0.

When training the fusion model, we use $\lambda_1 = \lambda_2 = 0.2$ and $k = 5$; that is, more attention is given to the FDB area because it is difficult to obtain.

IV. EXPERIMENTAL STUDIES

This section will present the dataset generation, implementation settings, comparison settings and experimental results.

A. Dataset Generation

A good training dataset should represent comprehensive situations of the task. The best choice for training data is using real photos. However, there are few multi-focus source images for fusion, and the ground truth needs to be labeled manually, which is very costly. Therefore, a feasible way is to generate artificial training images that are similar to reality yet easy to obtain. In our case, a dataset of foreground images with ground truth is used, and some images without an obvious defocus are chosen as the background dataset. Both the original foreground (FG^C) and the background (BG^C) images are first processed by Gaussian filters with kernel $G(x, y; \sigma)$ for the blurred images:

$$FG^B(x, y) = G(x, y; \sigma) \otimes FG^C(x, y), \quad (20)$$

$$BG^B(x, y) = G(x, y; \sigma) \otimes BG^C(x, y). \quad (21)$$

When the foreground is in focus, the ground truth is the same as the matte α^C , and when the background is in focus, the matte α^B is the blurred ground truth with the Gaussian kernel $G(x, y; \sigma)$:

$$\alpha^B(x, y) = G(x, y; \sigma) \otimes \alpha^C(x, y). \quad (22)$$

Then, the source images are generated using Equation (12) according to the matte (α^C or α^B) pixel by pixel. Here, we use source images 1 and 2 to denote the image pairs generated by the proposed model and source images A and B as the input of the network. Source image 1 ($ImgS_1$) has the in-focus foreground and the out-of-focus background, and source image 2 ($ImgS_2$) has the out-of-focus foreground and the in-focus background :

$$ImgS_1 = FG^C + (1 - \alpha^C)BG^B, \quad (23)$$

$$ImgS_2 = FG^B + (1 - \alpha^B)BG^C. \quad (24)$$

Examples of the source image pairs from the generated training dataset are also shown in Fig. 6(a) and Fig. 6(c). The defocus spread effect when the foreground is out of focus can be seen in the enlarged images in Fig. 6(d), and the comparison with the one when the foreground is in focus is shown in Fig. 6(b). Fusion of ground truth (Fig. 6(g)) is generated with the matte α^C :

$$GT = FG^C + (1 - \alpha^C)BG^C. \quad (25)$$

The guidance maps (Fig. 6(e) and Fig. 6(f)) are created at the same time. In the blurred matte α^B , the value in (0, 1) is set to 0.5 as the guidance map. The size of FDB area would be influenced by the level of defocus effect. Here 0.5 is used to indicate that the FDB area would be influenced by both foreground and background, which should be dealt with independently. If the foreground is in focus in source image A and out of focus in source image B, the guidance map $Gmap$ will be

$$Gmap(x, y) = \begin{cases} 0, & \alpha^B(x, y) = 0 \\ 0.5, & 0 < \alpha^B(x, y) < 1 \\ 1, & \alpha^B(x, y) = 1 \end{cases}. \quad (26)$$

On the other hand, if the foreground is defocused in source image A and focused in source image B, the guidance map $Gmap'$ will be the opposite of $Gmap$:

$$Gmap'(x, y) = \begin{cases} 1, & \alpha^B(x, y) = 0 \\ 0.5, & 0 < \alpha^B(x, y) < 1 \\ 0, & \alpha^B(x, y) = 1 \end{cases}. \quad (27)$$

We collect 200 foreground images from datasets of matting [1], [26] with corresponding matte maps, choosing 1,200 background pictures from the COCO dataset [27]. The background pictures are first resized to 512×512 . Then, for every foreground image, 20 background images are randomly chosen. The order of source images is random, with a probability equal to 0.5; therefore, 4,000 image pairs are obtained in total.

B. Training Settings

For the training process, $4 \times 1080Ti$ GPUs are used, and the test is carried out on a single GPU. The Adam solver is used with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The batch size is set to 32, with the learning rate set to 0.001. The model is trained on the generated dataset for 80 epochs. During the test process, it takes 0.27 seconds on average to fuse an image pair of size 520×520 .

C. Comparison Settings

We compare the proposed MMF-Net with 7 other multi-focus fusion methods: conventional methods including NSCT [5], SR [6], NSCT-SR [8], GF [12] and DSIFT [11]; network approaches including DCNN [15]; and our previous work BA-Fusion [21]. The experiments are conducted on the 'Lytro' [28] and 'Real-MFF' [29] datasets. 'Lytro' is commonly used for MFIF, and 'Real-MFF' is a new large MFIF dataset.

Four widely used objective metrics to assess fused image quality are used to evaluate the results [19]: average gradient (AG) [30], linear index of fuzziness (LIF) [31], mean square deviation (MSD) and gray level difference (GLD). Their formulations are described as follows.

1) LIF : LIF is an evaluation metric that can evaluate the enhancement of fused images:

$$LIF = \frac{2}{MN} \sum_{m=1}^M \sum_{n=1}^N \min\{p_{mn}, (1 - p_{mn})\}, \quad (28)$$

$$p_{mn} = \sin\left[\frac{\pi}{2} \left(1 - \frac{I(m, n)}{I_{max}}\right)\right], \quad (29)$$

where $I(m, n)$ is the intensity of pixel (m, n) in image I , and I_{max} is the maximum intensity of image I . A small LIF indicates that the enhancement of the fused image is good.

2) AG : AG is a metric that uses gradient information to measure the quality of fused images:

$$AG = \frac{1}{(M-1)(N-1)} \times \sum_{m=1}^{M-1} \sum_{n=1}^{N-1} \frac{1}{4} \sqrt{\left(\frac{\partial I(m, n)}{\partial m}\right)^2 + \left(\frac{\partial I(m, n)}{\partial n}\right)^2}, \quad (30)$$

where $\frac{\partial I(m, n)}{\partial m}$ and $\frac{\partial I(m, n)}{\partial n}$ are the gradients of the image in horizontal and vertical directions, respectively. A larger AG means that the boundaries of the fused image are clearer.

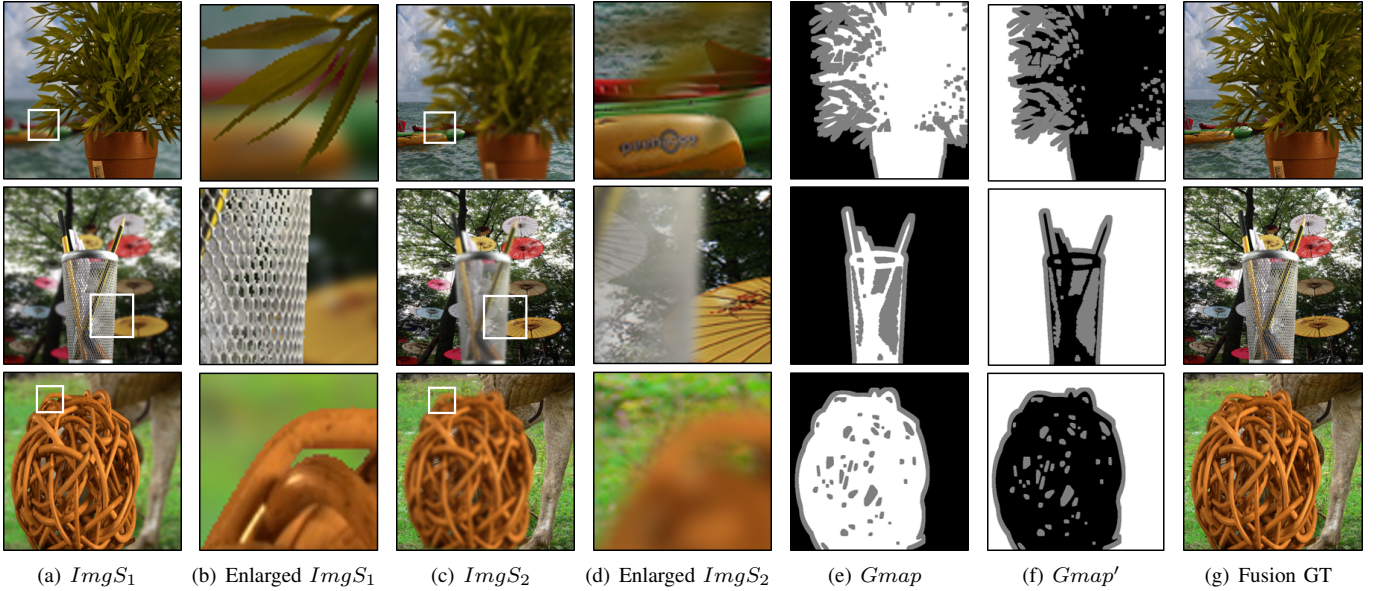


Fig. 6. Examples from the generated training dataset. The defocus spread effect is slight but can be seen in the enlarged images.

3) *MSD*: *MSD* measures image detail richness by calculating the difference between the intensity of each pixel and the mean intensity \bar{I} of the fused image:

$$MSD = \frac{1}{(M-1)(N-1)} \sqrt{\sum_{m=1}^{M-1} \sum_{n=1}^{N-1} (I(m,n) - \bar{I})^2}. \quad (31)$$

A larger *MSD* corresponds to a clearer fused image.

4) *GLD*: *GLD* uses the *L1* norm to calculate the gradient information of the fused image:

$$GLD = \frac{1}{(M-1)(N-1)} \times \sum_{m=1}^{M-1} \sum_{n=1}^{N-1} (|I(m,n) - I(m+1,n)| + |I(m,n) - I(m,n+1)|). \quad (32)$$

A larger *GLD* indicates a fused image with clearer boundary.

D. Experimental Results and Analysis

Fig. 7 shows the visual comparisons on the ‘Lytro’ dataset [28]. We select three image pairs to show the advantages of the proposed MMF-Net in three situations of different complexities: with small defocus areas, normal FDB and complex FDB. The top row (‘Lytro-01’) shows the ability of MMF-Net to handle small defocus areas. In source image A (Fig. 7(a)), the enlarged square is an out-of-focus area surrounded by the focused object. The proposed method (Fig. 7(j)) obtains a clearer grassland that is similar to the focused background in source B (Fig. 7(b)), especially compared with the spatial domain methods (Figs. 7(d), 7(f), 7(g), 7(h) and 7(i)).

The middle row (‘Lytro-11’) demonstrates the ability of MMF-Net to address normal FDB. In source image A (Fig. 7(a)), the enlarged square is focused on the finger and out-of-focus in the background. The transform domain

methods (Figs. 7(c) and 7(e)) fail in this situation, and the ghost effect exists around the boundary of the finger. In contrast, the proposed MMF-Net (Fig. 7(j)) obtains a clearer foreground similar to that in the source image B (Fig. 7(b)).

The bottom row (‘Lytro-05’) is an example in which MMF-Net contends with more complex FDB. The proposed method (Fig. 7(j)) produces clear results even in such a difficult situation. In the enlarged square, the artifact effect exists in the grille’s edge of Figs. 7(d), 7(g) and 7(i); unclear regions remain in Figs. 7(f) and 7(h); and defocus also exists in Figs. 7(c) and 7(e), as pointed out by the blue arrows. In the results of these methods, the textures in the sock and sole are different from that in Fig. 7(b), while our MMF-Net successfully preserves these textures in Fig. 7(j).

Fig. 8 shows the visual comparisons on the ‘Real-RFF’ dataset [29]. Some quite difficult fusion tasks are included in the dataset. As shown in Figs. 8(a) and 8(b), the foreground tree at the upper part is in focus in the source image B, and the background building is in focus in the source image A, while there are several falling leaves (the left enlarged square), which makes the situation complex. The proposed MMF-Net obtains a more satisfactory fusion result (Fig. 8(j)), compared with the SR, GF, DSIFT and CNN (Figs. 8(d), 8(f), 8(g) and 8(h)).

We also employ the difference map with the source image A to clearly show the comparison of different methods because the visual results are not always easy to distinguish. In our implementation,

$$DifferenceMap = k|Result - SourceA|, \quad (33)$$

and here we use $k = 15$ to illustrate the difference more clearly. Because the source image A is in focus in the background buildings and out of focus in the trees and falling leaves, the difference map with a pleasant fusion result would be all black in the building area (the right enlarged square) and with a clear difference on those falling leaves (the left enlarged square), which is the same as our results shown in

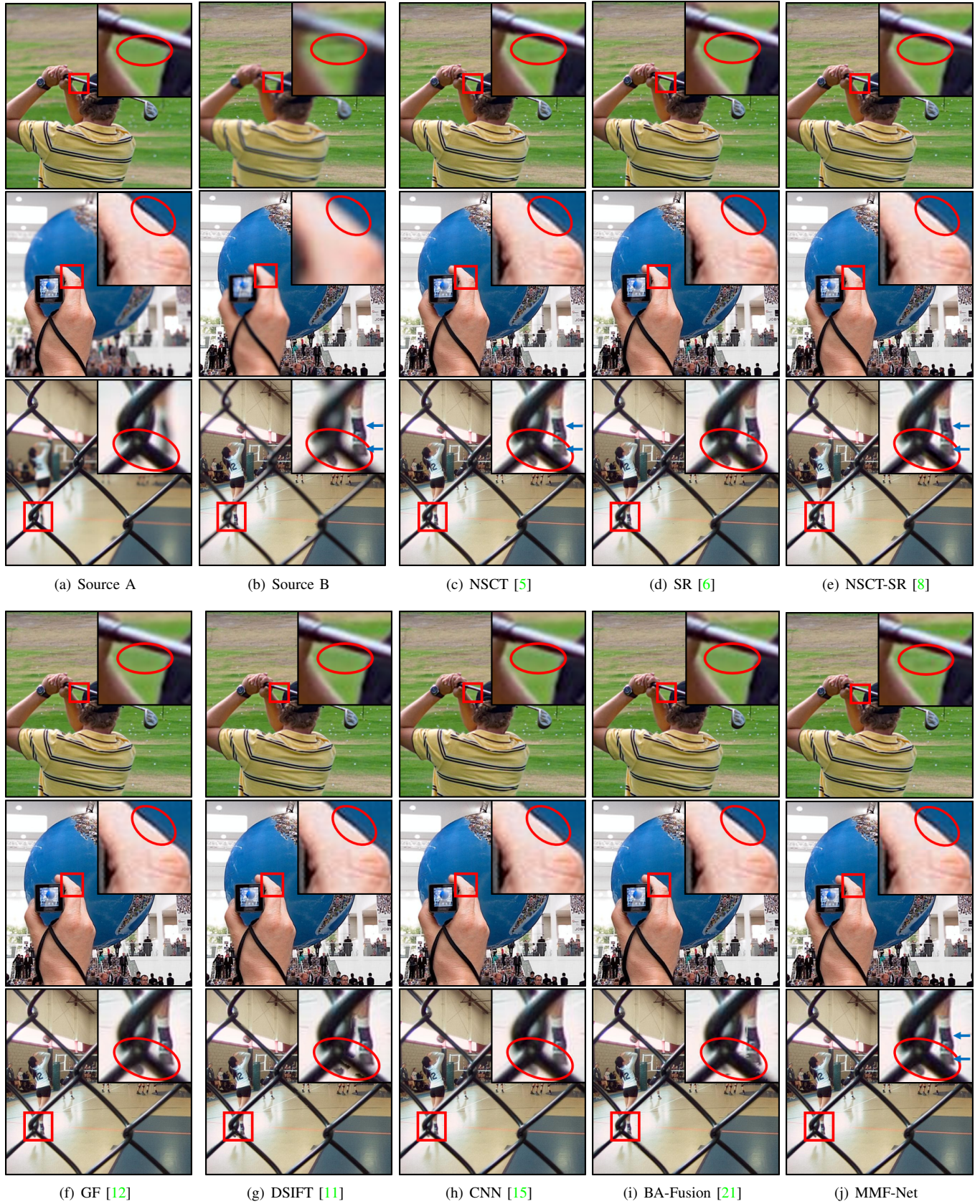


Fig. 7. The fusion results of different methods on the 'Lytro' dataset. Compared with other MFIF methods, both the edge of objects and background are clearer in the results of our MMF-Net, as shown in the enlarged squares.

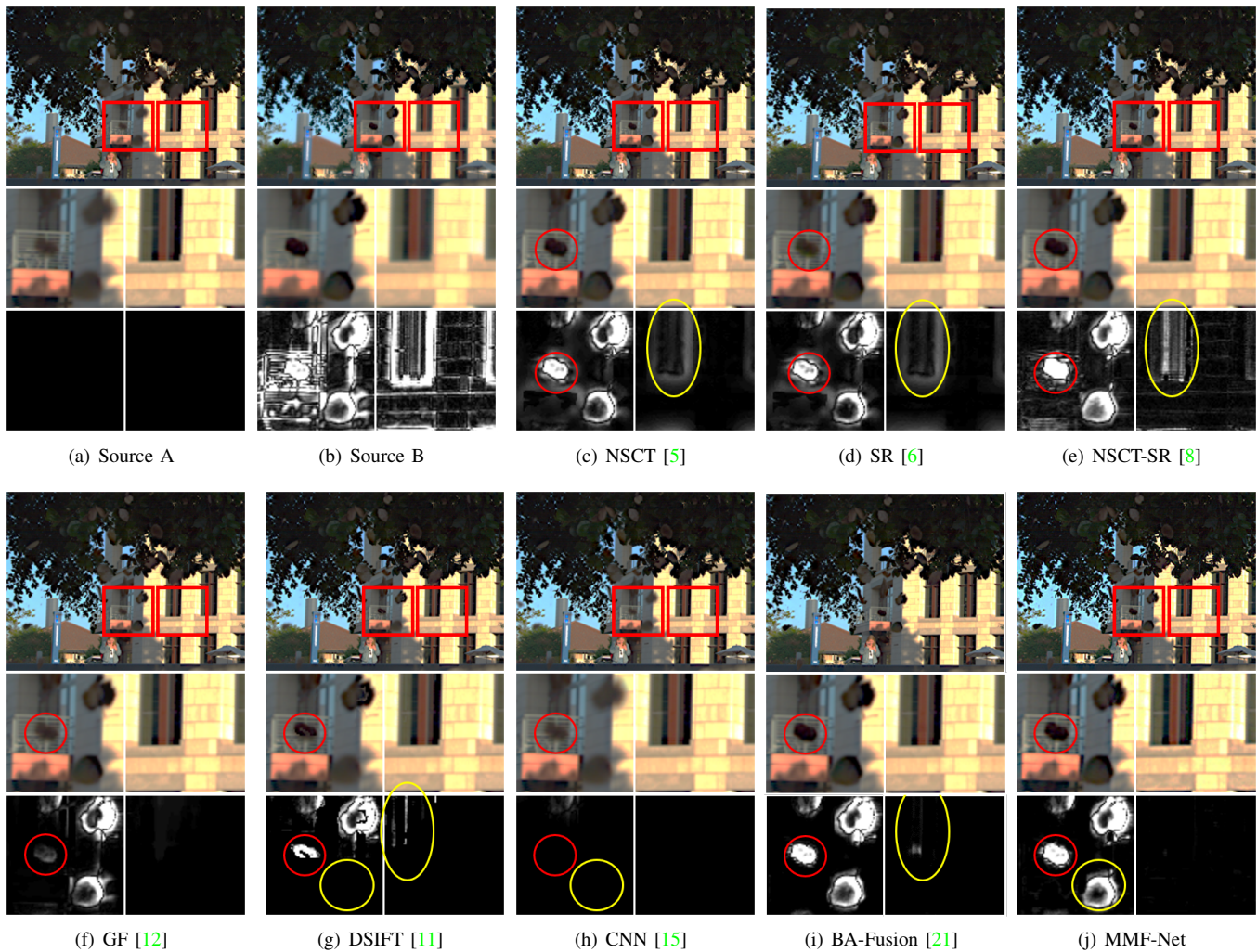


Fig. 8. The fusion results of different methods on the ‘Real-MFF’ dataset. Middle rows: Compared with other MFIF methods, our MMF-Net divides the foreground and background more precisely and obtains clearer results, as shown in the enlarged squares. Bottom rows: Difference maps with source image A are also provided here to show that our MMF-Net effectively preserves the information of in-focus area.

Fig. 8(j). In the buildings area, the MMF-Net keeps the clear results similar to source A, for which NSCT, SR, NSCT-SR, DSIFT and BA-Fusion (Figs. 8(c), 8(d), 8(e), 8(g) and 8(i)) fail. The missing of falling leaves in Figs. 8(d), 8(f), 8(g) and 8(h) can be noticed in the difference map as well.

The quantitative comparisons are shown in Table I and Table II. The larger values of metrics AG , MSD and GLD mean better fusion results, while the smaller values of LIF mean better results. The best results of the compared methods are highlighted in bold. The results in Table I show the average values over the 20 pairs of test images in the ‘Lytro’ dataset. Among the 20 pairs, the numbers of image pairs in which one method surpasses all the other methods are shown in the parentheses. As seen, the proposed MMF-Net remarkably outperforms the other fusion methods in terms of all quality metrics. We also conduct a comparison on the new ‘Real-MFF’ dataset [29], which includes 710 pairs of defocus images. The results are listed in Table II. All the image pairs are used for testing and, as the results show, the proposed MMF-Net outperforms the other fusion methods on all quality

metrics in this dataset as well.

E. Ablation Study

Several ablation studies are also conducted to show the effectiveness of the proposed defocus model and fusion network in Table III. All the ablation studies are carried out on the new Real-MMF dataset because it is much larger than the Lytro dataset.

To show the effectiveness of the proposed defocus model, we first establish two datasets based on the one-parameter (OnePara. Model) and two-parameter (TwoPara. Model) defocus models, respectively. Then, we train the proposed MMF-Net on these two datasets. To make the comparison convincing, same settings are used in the generation of data and the training process, including the size, resolution of the data, the number of epochs and the learning rate, etc. The results show that the network trained on the dataset generated with our α -matte boundary defocus model obtains a much better result on all the evaluation metrics. The two-parameter defocus model is no better than the one-parameter defocus model when used

TABLE I

THE QUANTITATIVE COMPARISON OF DIFFERENT MFIF METHODS ON THE LYTRO DATASET. FOR *AG*, *MSD* AND *GLD*, THE LARGER VALUES MEAN BETTER RESULTS; FOR *LIF*, THE SMALLER VALUES MEAN BETTER RESULTS. THE BEST AVERAGE RESULTS ARE IN BOLD, AND THE NUMBERS OF IMAGE PAIRS (OUT OF A TOTAL OF 20) IN WHICH ONE METHOD SURPASSES ALL THE OTHER METHODS ARE SHOWN IN THE PARENTHESES. OUR PROPOSED MMF-NET OUTPERFORMS THE OTHER MFIF METHODS ON ALL METRICS.

Metrics	NSCT [5]	SR [6]	NSCT-SR [8]	GF [12]	DSIFT [11]	DCNN [15]	BA-Fusion [21]	MMF-Net
<i>AG</i>	2.8750 (0)	2.8446 (0)	2.8794 (0)	2.8699 (0)	2.9020 (1)	2.8598 (0)	2.9040 (2)	2.9791 (17)
<i>MSD</i>	0.1108 (2)	0.1108 (0)	0.1108 (1)	0.1110 (1)	0.1111 (0)	0.1109 (0)	0.1112 (0)	0.1120 (16)
<i>GLD</i>	14.2245 (0)	14.0740 (0)	14.2467 (0)	14.1954 (0)	14.3400 (0)	14.1460 (0)	14.3550 (2)	14.7451 (18)
<i>LIF</i>	0.4097 (0)	0.4083 (0)	0.4093 (1)	0.4081 (1)	0.4075 (2)	0.4080 (0)	0.4075 (1)	0.4056 (15)

TABLE II

THE QUANTITATIVE COMPARISON OF DIFFERENT MFIF METHODS ON THE REAL-MFF DATASET. FOR *AG*, *MSD* AND *GLD*, THE LARGER VALUES MEAN BETTER RESULTS; FOR *LIF*, THE SMALLER VALUES MEAN BETTER RESULTS. THE BEST AVERAGE RESULTS ARE IN BOLD. OUR PROPOSED MMF-NET OUTPERFORMS THE OTHER MFIF METHODS ON ALL METRICS.

Metrics	NSCT [5]	SR [6]	NSCT-SR [8]	GF [12]	DSIFT [11]	DCNN [15]	BA-Fusion [21]	MMF-Net
<i>AG</i>	2.6660	2.5233	2.6670	2.6057	2.6077	2.6050	2.6158	3.5110
<i>MSD</i>	0.1000	0.0989	0.1000	0.0996	0.0997	0.0996	0.0998	0.1037
<i>GLD</i>	13.2039	12.5019	13.2089	12.8972	12.9063	12.8943	12.9471	17.4421
<i>LIF</i>	0.2643	0.2645	0.2640	0.2637	0.2642	0.2641	0.2634	0.2596

TABLE III

ABLATION STUDY. WE USE THE QUANTITATIVE COMPARISONS ON THE REAL-MFF DATASET WITH DIFFERENT TRAINING SETTINGS. FOR *AG*, *MSD* AND *GLD*, THE LARGER VALUES MEAN BETTER RESULTS; FOR *LIF*, THE SMALLER VALUES MEAN BETTER RESULTS. THE BEST AVERAGE RESULTS ARE IN BOLD. THE FULL MMF-NET OUTPERFORMS THE OTHER METHODS ON ALL METRICS WITH A REASONABLE TRAINING DATA GENERATION METHOD, FUNCTIONAL LOSS FUNCTION AND NETWORK.

Metrics	BA-Fusion [21]	OnePara. Model	TwoPara. Model	No $Loss_W$	No $Gmap$	Complete MMF-Net
<i>AG</i>	2.6158	2.8129	2.9331	2.8723	2.7143	3.5110
<i>MSD</i>	0.0998	0.1004	0.1010	0.1003	0.1001	0.1037
<i>GLD</i>	12.9471	13.9458	14.5308	14.2646	13.4515	17.4421
<i>LIF</i>	0.2634	0.2558	0.2618	0.2625	0.2628	0.2596

for data generation. Because the two-parameter model was first proposed for defocus estimation [8], it would have the anti-gradient effect on the boundary area. The results of BA-Fusion [21] are listed here as well because it is also trained with a one-parameter defocus model-based dataset; we can observe that the proposed MMF-Net trained on the one-parameter model also outperforms BA-Fusion.

Then, we conduct several experiments to show the effectiveness of the proposed MMF-Net. We first train the same network without the weighted loss ('No $Loss_W$ '), so no attention would be specifically paid to the boundary area. (Here, $\lambda_1 = 0.8$ and $\lambda_2 = 0.2$.) The results are slightly worse than the complete MMF-Net trained with $Loss_W$. Subsequently, we train the network without the $Gmap$ supervision ('No $Gmap$ ') and find that its results are worse than those of the 'No $Loss_W$ ' model, indicating the effectiveness of the applied guidance map.

V. CONCLUSIONS

In this paper, a cascaded boundary-aware convolutional network called MMF-Net is proposed for multi-focus image fusion along with a new α -matte boundary defocus model. The proposed MMF-Net aims to solve the unclear areas near the focused/defocused boundary (FDB) in the fusion results by implementing two subnets that first generate a fusion guidance map and then refine the fusion results in the areas near the FDB. In addition, the dataset generated with the α -matte model simulates the real-world images precisely, especially for the areas near the FDB. Experiments show that with the help

of MMF-Net and the more realistic training data, the proposed method outperforms the state-of-the-art ones both qualitatively and quantitatively.

In this work, we mainly propose the defocus spread model and then define a network that uses a relatively simple and direct architecture to illustrate the effectiveness of this model on multi-focus image fusion. Our future work will focus on the improvement of the network structure and using the attention mechanism [32], [33] to improve on the direct boundary guidance map.

REFERENCES

- [1] N. Xu, B. Price, S. Cohen, and T. Huang, "Deep image matting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2970–2979.
- [2] R. Hassen, Z. Wang, and M. M. Salama, "Objective quality assessment for multiexposure multifocus image fusion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2712–2724, 2015.
- [3] O. Bouzou, I. Andreadis, and N. Mitianoudis, "Conditional random field model for robust multi-focus image fusion," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5636–5648, 2019.
- [4] P. Kaur and M. Kaur, "A comparative study of various digital image fusion techniques: A review," *International Journal of Computer Applications*, vol. 114, no. 4, pp. 26–31, 2015.
- [5] Q. Zhang and B.-l. Guo, "Multifocus image fusion using the nonsub-sampled contourlet transform," *Signal processing*, vol. 89, no. 7, pp. 1334–1346, 2009.
- [6] B. Yang and S. Li, "Multifocus image fusion and restoration with sparse representation," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 4, pp. 884–892, 2010.
- [7] Q. Zhang and M. D. Levine, "Robust multi-focus image fusion using multi-task sparse representation and spatial context," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2045–2058, 2016.

- [8] S. Liu, F. Zhou, and Q. Liao, "Defocus map estimation from a single image based on two-parameter defocus model," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5943–5956, 2016.
- [9] M. Li, W. Cai, and Z. Tan, "A region-based multi-sensor image fusion scheme using pulse-coupled neural network," *Pattern Recognition Letters*, vol. 27, no. 16, pp. 1948–1956, 2006.
- [10] X. Qin, J. Shen, X. Mao, X. Li, and Y. Jia, "Robust match fusion using optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 8, pp. 1549–1560, 2014.
- [11] Y. Liu, S. Liu, and Z. Wang, "Multi-focus image fusion with dense SIFT," *Information Fusion*, vol. 23, pp. 139–155, 2015.
- [12] S. Li, X. Kang, and J. Hu, "Image fusion with guided filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2864–2875, 2013.
- [13] Z. Wang, Y. Ma, and J. Gu, "Multi-focus image fusion using PCNN," *Pattern Recognition*, vol. 43, no. 6, pp. 2003–2016, 2010.
- [14] H. Tang, B. Xiao, W. Li, and G. Wang, "Pixel convolutional neural network for multi-focus image fusion," *Information Sciences*, vol. 433–434, pp. 125–141, 2018.
- [15] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Information Fusion*, vol. 36, pp. 191–207, 2017.
- [16] X. Guo, R. Nie, J. Cao, D. Zhou, L. Mei, and K. He, "FuseGAN: Learning to fuse multi-focus image via conditional generative adversarial network," *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 1982–1996, 2019.
- [17] B. Ma, X. Ban, H. Huang, and Y. Zhu, "SESF-Fuse: An unsupervised deep model for multi-focus image fusion," *arXiv preprint arXiv:1908.01703*, 2019.
- [18] X. Yan, S. Z. Gilani, H. Qin, and A. Mian, "Unsupervised deep multi-focus image fusion," *arXiv preprint arXiv:1806.07272*, 2018.
- [19] W. Zhao, D. Wang, and H. Lu, "Multi-focus image fusion with a natural enhancement via a joint multi-level deeply supervised convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1102–1115, 2018.
- [20] S. Zhuo and T. Sim, "Defocus map estimation from a single image," *Pattern Recognition*, vol. 44, no. 9, pp. 1852–1858, 2011.
- [21] H. Ma, J. Zhang, S. Liu, and Q. Liao, "Boundary aware multi-focus image fusion using deep neural network," in *IEEE International Conference on Multimedia and Expo*, 2019, pp. 1150–1155.
- [22] K. He, J. Sun, and X. Tang, "Guided image filtering," in *European Conference on Computer Vision*. Springer, 2010, pp. 1–14.
- [23] Y. Chen, J. Guan, and W.-K. Cham, "Robust multi-focus image fusion using edge model and multi-matting," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1526–1541, 2017.
- [24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [26] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, "A perceptually motivated online benchmark for image matting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1826–1833.
- [27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, 2014, pp. 740–755.
- [28] M. Nejadi, S. Samavi, and S. Shirani, "Multi-focus image fusion using dictionary-based sparse representation," *Information Fusion*, vol. 25, pp. 72–84, 2015.
- [29] J. Zhang, Q. Liao, S. Liu, H. Ma, W. Yang, and J.-H. Xue, "Real-MFF: A large realistic multi-focus image dataset with ground truth," *Pattern Recognition Letters*, vol. 138, pp. 370–377, 2020.
- [30] G. Cui, H. Feng, Z. Xu, Q. Li, and Y. Chen, "Detail preserved fusion of visible and infrared images using regional saliency extraction and multi-scale image decomposition," *Optics Communications*, vol. 341, pp. 199–209, 2015.
- [31] X. Bai, F. Zhou, and B. Xue, "Noise-suppressed image enhancement using multiscale top-hat selection transform through region extraction," *Applied optics*, vol. 51, no. 3, pp. 338–347, 2012.
- [32] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2368–2378, 2017.
- [33] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1531–1544, 2018.



Haoyu Ma received the B.S. and M.Eng. degrees from the Department of Electronic Engineering, Tsinghua University, China, in 2017 and 2020. His research interests include defocus imaging, image fusion and stereo vision.



Qingmin Liao received the B.S. degree in radio technology from the University of Electronic Science and Technology of China, China, in 1984, and the M.S. and Ph.D. degrees in signal processing and telecommunications from the University of Rennes 1, France, in 1990 and 1994. He is a Professor in the Department of Electronic Engineering, Tsinghua University. His research interests include image/video processing, transmission and analysis; biometrics; and their applications to teledetection, medicine, industry, and sports.



Juncheng Zhang received the B.S. degree from the College of Electronics and Information Engineering, Sichuan University, China, in 2016. He is currently pursuing the Ph.D. degree in electronic engineering, Tsinghua University, China. His research interests include multi-focus image fusion and defocus blur identification.



Shaojun Liu received the B.S. and Ph.D. degrees from the Department of Electronic Engineering, Tsinghua University, China, in 2014 and 2019. He is currently a postdoctoral fellow with the Department of Electronic and Computer Engineering at Hong Kong University of Science and Technology, Hong Kong, China. His research interests include defocus blur identification, multi-focus image fusion and medical image analysis.



Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. He is an Associate Professor in the Department of Statistical Science at University College London. His research interests include statistical classification, high-dimensional data analysis, computer vision and pattern recognition.