Towards Reducing Severe Defocus Spread Effects for Multi-Focus Image Fusion via an Optimization Based Strategy

Shuang Xu, Lizhen Ji, Zhe Wang, Pengfei Li, Kai Sun, Chunxia Zhang, and Jiangshe Zhang

***Abstract*—Multi-focus image fusion (MFF) is a popular technique to generate an all-in-focus image, where all objects in the scene are sharp. However, existing methods pay little attention to defocus spread effects of the real-world multi-focus images. Consequently, most of the methods perform badly in the areas near focus map boundaries. According to the idea that each local region in the fused image should be similar to the sharpest one among source images, this paper presents an optimization-based approach to reduce defocus spread effects. Firstly, a new MFF assessment metric is presented by combining the principle of structure similarity and detected focus maps. Then, MFF problem is cast into maximizing this metric. The optimization is solved by gradient ascent. Experiments conducted on the real-world dataset verify superiority of the proposed model. The codes are available at https://github.com/xsxjtu/MFF-SSIM.**

***Index Terms*—Multi-focus image fusion, defocus spread effect, structure similarity.**

I. Introduction

DUE to the limitation of imaging devices and their depth-of-field operation, it is hard to acquire all-in-focus images [1]. In general, only one plane scene stays in focus and others not in focus are blurred. Multi-focus image fusion (MFF) is a useful and promising digital image post-processing technique to cope with this problem. It generates an all-in-focus image by integrating complementary information from source images of the same scene taken at different focus distances.

Manuscript received June 14, 2020; revised August 16, 2020 and October 11, 2020; accepted November 17, 2020. Date of publication November 24, 2020; date of current version December 14, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102201, and in part by the National Natural Science Foundation of China under Grants 11671317, 61976174. The associate editor coordinating the review of this manuscript and approving it for publication was J. Gu. *(Corresponding author: Jiangshe Zhang.)*

Shuang Xu, Lizhen Ji, Kai Sun, Chunxia Zhang, and Jiangshe Zhang are with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 12480, China (e-mail: shuangxu@stu.xjtu.edu.cn; jlz_stat@stu.xjtu.edu.cn; kaisun@mail.xjtu.edu.cn; cxzhang@mail.xjtu.edu.cn; jszhang@mail.xjtu.edu.cn).

Zhe Wang is with the Department of Computer Science, University of Virginia, Charlottesville, VA 22904 USA (e-mail: zw6sg@virginia.edu).

Pengfei Li is with the Universal Text group, the Department of Intelligent Traffic, Hikvision, Shanghai 201100, China (e-mail: lipengfei27@hikvision.com).

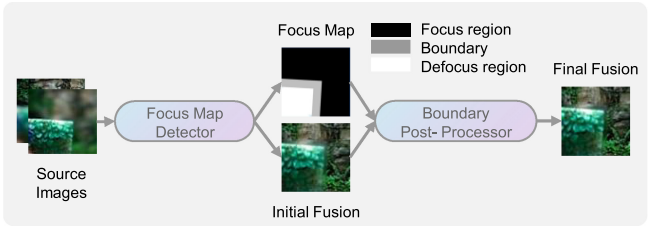
This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2020.3039564, provided by the authors. The material includes the results on Lytro and Grayscale datasets. This material is 3.17MB in size.

Digital Object Identifier 10.1109/TCI.2020.3039564

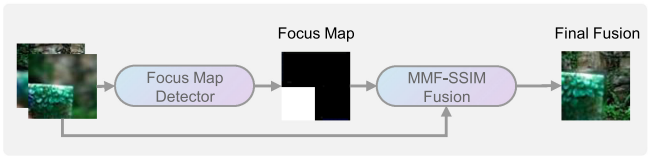
The existing methods can be classified into two groups. The first one is the transform domain based methods. Its basic idea is to utilize a transformer (e.g., discrete Fourier transform[2], discrete wavelet transform [3], non-subsampled contourlet transform [4], etc.) to convert source images into the feature domain, in which salient features can be easily detected. The fused image is reconstructed from feature domain to spatial domain after merging salient features according to a certain fusion strategy. However, it is reported that transform domain based methods tend to result in the brightness or color distortion because they do not take spatial consistency into account [5]. With the development of dictionary learning [6], [7], sparse representation based image fusion has emerged as a special transform domain method [8]. Sparse representation outperforms classic transforms for its stability and robustness to noise and misregistration [9], [10]. Nonetheless, some details may be lost. Recently, Liu *et al.* present a general image fusion framework by means of integrating multiscale transform and sparse representation [11], which is able to simultaneously overcome their inherent shortcomings.

Spatial domain based methods belong to the second group. They detect the focus map and fuse images in the spatial domain. Generally speaking, how to define a focus measurement and how to accurately detect focus map play significant roles [12]. To obtain satisfactory results, various sophisticated methods that incorporate certain prior knowledge have been proposed. For example, with the aim at preserving salient edges and local shapes, Li *et al.* apply a guided filter [13] to decompose source images into base and detail layers [5]. Then, the base and detail layers are fused separately by means of weighted average strategy, where weights are represented by the detected focus map. Finally, the sharp image is reconstructed by combining fused base and detail layers. Li *et al.* employ the morphological filter to generate initial boundary between focus and defocus regions, and refine it by the matting technique [14]. However, these methods may lose efficiency when the detected focus map is inaccurate.

Recently, deep neural networks have emerged as effective tools for the MFF task. Liu *et al.* make the first attempt to apply a convolutional neural network (CNN) to detecting the focus map [15]. Then, they propose a framework for the general image fusion problem [16]. To deal with complicated focus maps, Li *et al.* design a novel network in the deep regression pair learning (DRPL) fashion [17]. Amin-Naji *et al.* ensemble the deep



(a) MMFNet



(b) Our framework

(c) Fusion image of MMFNet

(d) Fusion image of our framework

Fig. 1. The differences between MMFNet and our framework. Better view in electronic version.

features of three neural networks (ECNN) to obtain more accuracy results [18]. Nian and Jung develop a novel CNN to combine the light field data with multi-focus images [19]. The comprehensive comparison of these methods is reported in the recent surveys [20], [21]. Although deep learning based methods are powerful to learn a specific pattern, it is worth pointing out that most of the methods omit defocus spread effects of the real-world multi-focus images [21]–[24]. Generally speaking, the out-of-focus objects tend to expand. Hence, when the background object is in focus, the expanded foreground object will overlay the boundary between background and foreground. As a result, many methods are very likely to make mistakes around focus map boundaries and generate unrealistic images, if source images suffer from severe defocus spread effects [22]–[24]. Although several deep learning based methods have been proposed, most of them are devoted to improving the accuracy of focus maps instead of solving the defocus spread effect.

To the best of our knowledge, there are only two deep networks taking it into account. In references [22] and [23], they separately proposed two kinds of defocus models to generate synthetic images with defocus spread effects. In addition, they built and trained end-to-end deep neural networks on these synthetic models. The MMFNet proposed in reference [22] is the state-of-the-art (SOTA) method. As shown in Fig. 1(a), it consists of two sub-networks. The first sub-network serves as a focus map detector who segments the scene into three parts, including a focus region, a defocus region and a focus/defocus boundary. The second sub-network serves as a post-processing network who aims at enhancing the focus/defocus boundary. Nonetheless, the prerequisite of MMFNet performing well is that the focus map detector is accurate enough. Fig. 1(c) exhibits an example, where MMFNet mistakenly detects the focus map and fails to generate clear background. The weak generalization ability and non-robustness limit the application of MMFNet to real-world images.

To develop a robust fusion strategy, we present a novel optimization-based framework to solve defocus spread effects. The basic idea is to abandon pixel-wise fusion, and to make each local patch of the fused image similar to the corresponding region of the sharpest source image. One of the most suitable image quality metrics is structural similarity (SSIM) [25], [26], which evaluates the similarity between two images according to the luminance, structure and contrast. However, SSIM cannot be applied to MFF task, because we aim to evaluate the similarity between a fused image and a set of source images rather than a single reference image. To eliminate this obstacle, by combining detected focus maps and the principle of SSIM, we propose the multi-focus image fusion structural similarity (MFF-SSIM) index. Then, MFF-SSIM is taken as an objective function to search for a satisfactory result in the image space. Because MFF-SSIM index is a highly non-linear and non-convex function, it is hard to obtain an analytic solution. As an alternative, the gradient ascent algorithm is employed. Our contributions can be summarized as follows:

1) This paper proposes a novel metric called MFF-SSIM index, and the MFF task is turned into maximizing it. An iterative solution of this optimization problem is provided.
2) A series of experiments are conducted to demonstrate the superiority of MFF-SSIM model. It is revealed that compared with the SOTA techniques, our method is effective to tackle the defocus spread effects which occurs at depth edges in the image in the presence of severe defocus.

The rest of this paper is organized as follows. In Section II, we present MFF-SSIM index and introduce how to solve our model. Extensive experiments are conducted in Section III. At last, Section IV concludes the paper.

II. Model Formulation

To begin with, we introduce the notations in this paper. We use the calligraphy letter $\mathcal{X} \in R^{M\times N\times C}$ and the uppercase letter $\boldsymbol{X} \in R^{MNC}$ to denote the tensor and vector version of an image, respectively. Here, M, N and C denote the height, width and the number of channels, respectively. The lowercase letter $\boldsymbol{x}_i \in R^{CW^2}(i = 1, \ldots, P)$ denotes the local patch with window size W. The K preregistered multi-focus images and the fused image are denoted by $\{\boldsymbol{X}^{[k]}\}_{k=1}^{K}$ and $\boldsymbol{Y}$, respectively. Typically, $\boldsymbol{x}_i^{[k]}$ stands for the i^{th} patch of the k^{th} source image.

A. Defocus Spread Effects

As shown in Fig. 2, the defocus spread effect is a common phenomenon that the objects not in focus tend to expand. It brings two challenges for multi-focus image fusion algorithms:

1) Some objects not in focus will significantly expand, and they can confuse the focus map detector. As a result, the detection results are inaccurate, and the fused images contain artifacts or inconsistent contents.

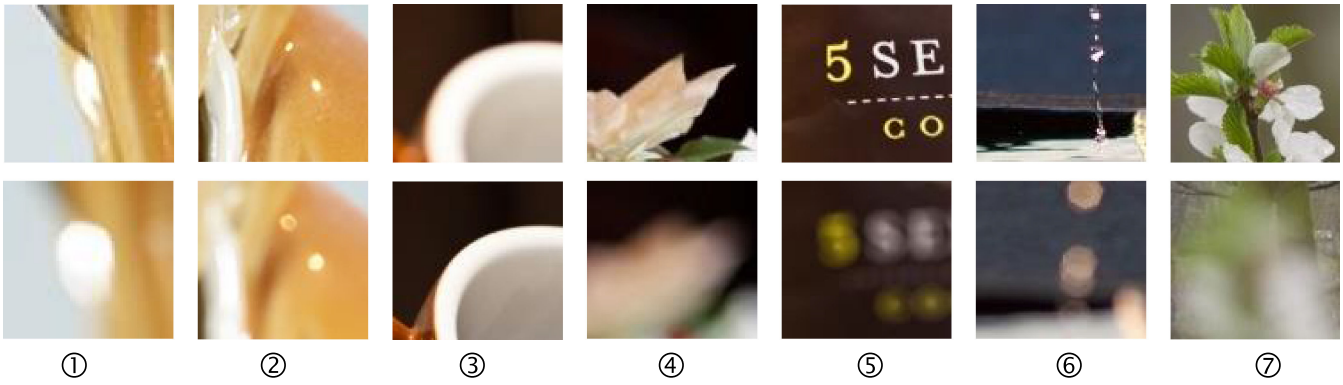


Fig. 2. Seven image pairs suffering from defocus spread effects.

2) Obviously, the blurred foreground will cover a part of clear background, when the background is in focus. On the other hand, the blurred background does not affect the clear foreground, when the foreground is in focus. Thus, there is a region between foreground and background being blurred whenever background or foreground is in focus (e.g., the 6th and the 7th image pairs in Fig. 2).

B. Motivation

Recently, segmentation-based methods have emerged as popular tools for MFF task. The basic idea is to detect a focus map $\mathcal{M}$ and then use addition strategy to generate fusion images by the following equation

$$\mathcal{Y} = \mathcal{M} \odot \mathcal{X}^{[1]} + (1 - \mathcal{M}) \odot \mathcal{X}^{[2]}, \tag{1}$$

where $\odot$ is the element-wise product and $\mathcal{M}$ is binary (its elements equal to 1 if it is in focus and 0 otherwise). This strategy does not consider defocus spread effects into account. To deal with this effect, MMFNet not only generates a focus map $\mathcal{M}$ but also estimate a focus/defocus boundary map $\mathcal{B}$ whose entries indicate whether the pixels are located at the focus/defocus boundary. In other words, MMFNet segments the scene into three parts, i.e., a focus region, a defocus region and the boundary. The fused image is computed by

$$\begin{aligned}\mathcal{Y} = (1 - \mathcal{B}) \odot \left[\mathcal{M} \odot \mathcal{X}^{[1]} + (1 - \mathcal{M}) \odot \mathcal{X}^{[2]}\right] \\ + \mathcal{B} \odot f(\mathcal{X}^{[1]}, \mathcal{X}^{[2]}),\end{aligned} \tag{2}$$

where $f(\cdot, \cdot)$ is a boundary post-processing function. However, we found that these two strategies would generate unsatisfactory results if focus maps are inaccuracy or source images suffer from severe defocus spread effects.

In order to reduce the sensitivity to detected focus maps and deal with the defocus spread effect, we present a new framework for the MFF task. It aims at fusing images patch-wise rather than pixel-wise. In intuition, patch-wise fusion may lead to more robust results. Taking the second pair of images in Fig. 2 as an example, all the pixels in the top patch are in focus. If the detector generates an inaccurate focus map as shown in Fig. 3, the pixel-wise methods would lead to artifacts. Even though these inaccurate maps can be refined by some morphology filters [15],

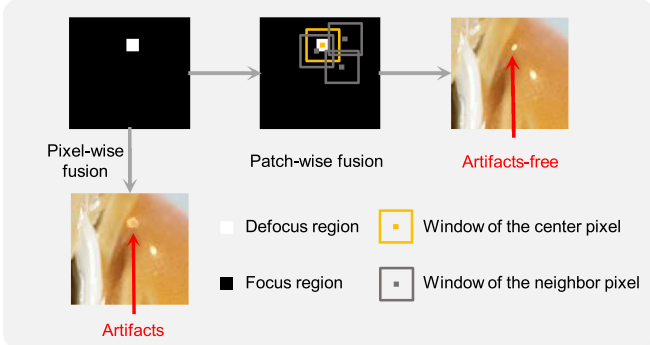


Fig. 3. The difference between pixel-wise and patch-wise fusion.

the experiments reported in Subsection III-B demonstrate that it still cannot meet the demand (see Fig. 6(b)). The patch-wise method fuses the source images in the patch level. Therefore, when a pixel is mistakenly detected and most of the neighbors in its local patch are correctly detected, this pixel can be corrected by the overlapped windows of its neighbors. According to this idea and inspired by [26], [27], we propose a novel patch-wise metric to assess the similarity between a fused image and a set of source images in Subsection II-C. Then, we regard it as the objective function and directly search for the optimal fusion image in the image space. At last, an efficient algorithm is presented to solve this optimization problem in Subsection II-D.

C. MFF-SSIM Index

SSIM is a widely used image quality metric. Given two image patches $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, SSIM [25] is defined by

$$\mathrm{SSIM}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \frac{a_1}{b_1}\frac{a_2}{b_2}, \tag{3}$$

where

$$\begin{aligned} a_1 &= 2\mu_{\boldsymbol{x}_i}\mu_{\boldsymbol{y}_i} + C_1, \quad & b_1 &= \mu_{\boldsymbol{x}_i}^2 + \mu_{\boldsymbol{y}_i}^2 + C_1, \\ a_2 &= 2\sigma_{\boldsymbol{x}_i\boldsymbol{y}_i} + C_2, \quad & b_2 &= \sigma_{\boldsymbol{x}_i}^2 + \sigma_{\boldsymbol{y}_i}^2 + C_2. \end{aligned} \tag{4}$$

Note that $\mu_{\boldsymbol{x}_i}$, $\sigma_{\boldsymbol{x}_i}^2$ and $\sigma_{\boldsymbol{x}_i\boldsymbol{y}_i}$ denote the mean, variance and covariance, respectively. C_1 and C_2 are small constants for numerical stability. Then, the final SSIM score for the two

images $\boldsymbol{X}$ and $\boldsymbol{Y}$ is averaged over all the patches, $Q(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{P}\sum_{i=1}^{P}\mathrm{SSIM}(\boldsymbol{x}_i,\boldsymbol{y}_i)$, where P denotes the number of patches. SSIM evaluates the similarity between two images, but it unfortunately cannot be directly applied to the MFF task, where we need to assess how much the information is transferred from a set of source images $\{\boldsymbol{X}^{[k]}\}_{k=1}^{K}$ to the fused image $\boldsymbol{Y}$.

To cope with this problem, we introduce the MFF-SSIM index. Generally speaking, a desired fusion image should incorporate the sharper regions of all source images. Hence, MFF-SSIM for the i^{th} patch is defined by

$$S(\{\boldsymbol{x}_i^{[k]}\}_{k=1}^{K},\boldsymbol{y}_i) = \mathrm{SSIM}(\boldsymbol{x}_i^{[j]},\boldsymbol{y}_i), \tag{5}$$

if the i^{th} patch of the j^{th} source image is sharpest, where $\boldsymbol{x}_i^{[k]}$ denotes the i^{th} patch of the k^{th} source image. In this fashion, MFF-SSIM is able to compare the fusion image with the sharpest one patch-to-patch. Given the detected focus map, Eq. (5) becomes

$$S(\{\boldsymbol{x}_i^{[k]}\}_{k=1}^{K},\boldsymbol{y}_i) = \sum_{k=1}^{K} m_{ik}\mathrm{SSIM}(\boldsymbol{x}_i^{[k]},\boldsymbol{y}_i), \tag{6}$$

where $m_{ik}\in\{0,1\}$ indicates whether the k^{th} source image is sharpest with regard to the i^{th} local patch, i.e., the focus map. Then, the final MFF-SSIM is obtained by averaging the local scores, i.e.,

$$Q(\{\boldsymbol{X}^{[k]}\}_{k=1}^{K},\boldsymbol{Y}) = \frac{1}{P}\sum_{i=1}^{P} S(\{\boldsymbol{R}_i\boldsymbol{X}^{[k]}\}_{k=1}^{K},\boldsymbol{R}_i\boldsymbol{Y}). \tag{7}$$

Note that the binary matrix $\boldsymbol{R}_i\in\{0,1\}^{CW^2\times MNC}$ is the patch extractor such that $\boldsymbol{R}_i\boldsymbol{X} = \boldsymbol{x}_i$.

D. MFF-SSIM Framework

Our motivation is to make each local patch of the fused image similar to the corresponding region of the sharpest source image. To this end, the MFF-SSIM index has been proposed to evaluate quality of the fused image in this sense. Therefore, the MFF task can be formulated as the following optimization problem, that is,

$$\max_{\boldsymbol{Y}} Q(\{\boldsymbol{X}^{[k]}\}_{k=1}^{K},\boldsymbol{Y}). \tag{8}$$

Owing to the high non-linearity and non-convexity of MFF-SSIM index, obtaining an analytic solution is a challenge. Consequently, we exploit the gradient ascent algorithm to solve this problem. Briefly speaking, given a current estimation of the fused image $\boldsymbol{Y}^{(t)}$ and a proper learning rate $\beta>0$, the update

$$\boldsymbol{Y}^{(t+1)} = \boldsymbol{Y}^{(t)} + \beta\boldsymbol{G}^{(t)} \tag{9}$$

will make objective function (MFF-SSIM index) increase. Note that $\boldsymbol{G}^{(t)}$ denotes the gradient with regard to (w.r.t.) $\boldsymbol{Y}^{(t)}$, $\nabla_{\boldsymbol{Y}}Q(\{\boldsymbol{X}^{[k]}\}_{k=1}^{K},\boldsymbol{Y})|_{\boldsymbol{Y}^{(t)}}$. It is easy to see that

$$\boldsymbol{G}^{(t)} = \frac{1}{P}\sum_{i=1}^{P}\boldsymbol{R}_i^{\mathrm{T}}\nabla_{\boldsymbol{Y}}S(\{\boldsymbol{R}_i\boldsymbol{X}^{[k]}\}_{k=1}^{K},\boldsymbol{R}_i\boldsymbol{Y}), \tag{10}$$

where $\boldsymbol{R}_i^T$ denotes the inverse patch extractor to place the gradient patch back into the corresponding entries of the original image. Therefore, the original problem is cast into the computation of gradient for a local patch. We have

$$\nabla_{\boldsymbol{y}_i}\mathrm{SSIM}(\boldsymbol{x}_i,\boldsymbol{y}_i) = \frac{a_2\nabla_{\boldsymbol{y}_i}a_1 + a_1\nabla_{\boldsymbol{y}_i}a_2}{b_1b_2} - \frac{a_1a_2(b_2\nabla_{\boldsymbol{y}_i}b_1 + b_1\nabla_{\boldsymbol{y}_i}b_2)}{(b_1b_2)^2}, \tag{11}$$

where

$$\begin{aligned}\nabla_{\boldsymbol{y}_i}a_1 &= 2\mu_{\boldsymbol{x}_i}\nabla_{\boldsymbol{y}_i}\mu_{\boldsymbol{y}_i} \quad ,\nabla_{\boldsymbol{y}_i}a_2 = 2\nabla_{\boldsymbol{y}_i}\sigma_{\boldsymbol{x}_i\boldsymbol{y}_i},\\ \nabla_{\boldsymbol{y}_i}b_1 &= 2\nabla_{\boldsymbol{y}_i}\mu_{\boldsymbol{y}_i} \qquad\quad ,\nabla_{\boldsymbol{y}_i}b_2 = \nabla_{\boldsymbol{y}_i}\sigma_{\boldsymbol{y}_i}^2.\end{aligned} \tag{12}$$

The gradient of mean, variance and covariance w.r.t. patch $\boldsymbol{y}_i$ are as follows,

$$\begin{aligned}\nabla_{\boldsymbol{y}_i}\mu_{\boldsymbol{y}_i} &= \frac{\mathbf{1}}{CW^2},\\ \nabla_{\boldsymbol{y}_i}\sigma_{\boldsymbol{y}_i}^2 &= \frac{2(\boldsymbol{y}_i-\boldsymbol{\mu}_{\boldsymbol{y}_i})}{CW^2},\\ \nabla_{\boldsymbol{y}_i}\sigma_{\boldsymbol{x}_i\boldsymbol{y}_i} &= \frac{(\boldsymbol{x}_i-\boldsymbol{\mu}_{\boldsymbol{x}_i})}{CW^2}.\end{aligned} \tag{13}$$

Here, $\mathbf{1}$ represents the vector whose all entries are one. According to the above equations, it is easy to write the gradient of a local patch as shown in Eq. (14).

$$\begin{aligned}&\nabla_{\boldsymbol{y}_i}S(\{\boldsymbol{x}_i^{[k]}\}_{k=1}^{K},\boldsymbol{y}_i)\\ &= \frac{2}{CW^2}\sum_{k=1}^{K} m_{ik}\left\{\frac{\mu_{\boldsymbol{x}_i^{[k]}}a_2\mathbf{1} + a_1(\boldsymbol{x}_i^{[k]}-\mu_{\boldsymbol{x}_i^{[k]}})}{b_1b_2} - \frac{a_1a_2[\mu_{\boldsymbol{y}_i}b_2\mathbf{1} + b_1(\boldsymbol{y_i}-\mu_{\boldsymbol{y}_i})]}{b_1^2b_2^2}\right\}.\end{aligned} \tag{14}$$

Recall that $m_{ik}\in\{0,1\}$ indicates whether the i^{th} pixel in k^{th} source image is in focus or not.

E. Implementation Details

Based on the above analysis, we summarize the main steps of MFF-SSIM model as shown in Algorithm 1. In our experiments, the hyper-parameter configuration is set as follows. When computing the MFF-SSIM index, we set $C_1 = 0.01^2$ and $C_2 = 0.03^2$. Furthermore, the overlapped patches are extracted with a stride of 1 to prevent from artifacts around patch boundaries. As for the window size, in Section III-B it is empirically set as $W = 5\times10^{-5}MN$. We also investigate how window size affects our algorithm in Section III-D. When we optimize MFF-SSIM, the learning rate β is set to 10^{-3}. Our optimization algorithm stops if the number of iterations exceeds 1000. The initial value is set by the average image, that is, $\boldsymbol{Y}^{(0)} = \sum_{k=1}^{K}\boldsymbol{X}^{[k]}/K$. We rescale images into [0,1]. At last, it should be emphasized that this algorithm needs a pre-detected focus map $\boldsymbol{M}$. In the next subsection, we propose two focus map detectors.

The computational complexity of our algorithm is $O(W^2P)$, where P is the number of local patches in an image. Large window size makes our algorithm slow. For a pair of 624×432

Algorithm 1: Gradient Ascent for MFF-SSIM Model.

Input: Initial fused image $\boldsymbol{Y}^{(0)}$, learning rate β, window size W, the maximum number of iterations T, small constants C_1, C_2, the detected focus map $\boldsymbol{M}$.
Output: Final fused image $\boldsymbol{Y}^{(t)}$
1: Compute MFF-SSIM value $Q^{(0)}$ and gradient $\boldsymbol{G}^{(0)}$ w.r.t. $\boldsymbol{Y}^{(0)}$;
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Update fused image $\boldsymbol{Y}^{(t)}$ by eqs. (9) and (14);
4: Compute MFF-SSIM value $Q^{(t)}$ and gradient $\boldsymbol{G}^{(t)}$ w.r.t. $\boldsymbol{Y}^{(t)}$;
5: **end for**

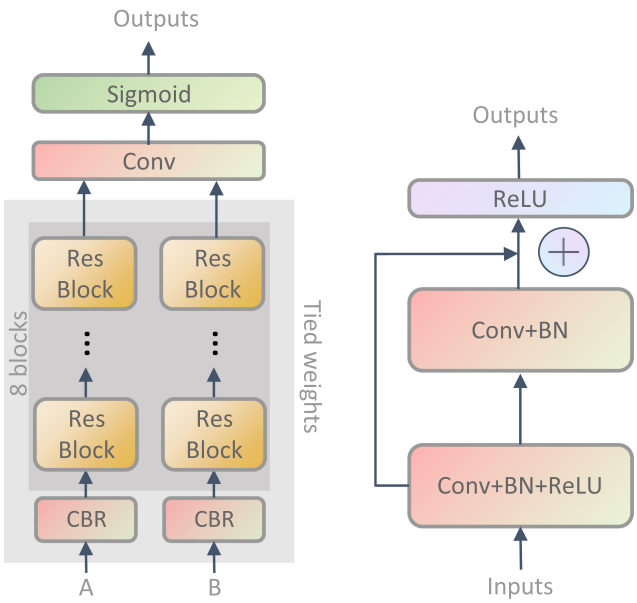


Fig. 4. Left: The structure of a residual network. Right: The residual block.

color images, it takes around 3.23s per iteration with an Intel Core i7-8750H CPU at 2.20GHz.

F. Focus Map Detectors

By now, we have proposed the MFF-SSIM framework, but it still lacks a focus map detector. In this paper, we provide two detectors, that is, the Laplacian energy and a residual network.

According to the fact that a sharp/blurred image has greater/smaller gradients, the focus map can be determined by the gradient intensity. Let $\mathcal{L}$ denote the Laplacian filter, and the gradient intensity of a local patch $\boldsymbol{x}_i$ can be quantified by the Laplacian energy, which is defined by

$$e_{\boldsymbol{x}_i} = \sum_{m,n} (\mathcal{L}(\boldsymbol{x}_i))^2_{mn}. \tag{15}$$

In this way, for the focus map $\boldsymbol{M}$, we have $m_{ik} = 1$ if $e_{\boldsymbol{x}_i^{[k]}}$ is largest among $\{\boldsymbol{x}_i^{[k]}\}_{k=1}^K$ and 0 otherwise.

The second detector employs a deep convolutional network to estimate the focus map $\boldsymbol{M}$. The network structure is displayed in Fig. 4. At first, the network starts with the CBR (that is, a **c**onvolutional unit, a **b**atch normalization (BN) layer and a **r**ectified linear unit (ReLU)) and 8 residual blocks to separately extract initial feature maps for images A and B. Two feature maps are concatenated along channels and then are put in a convolutional unit and a sigmoid function to generate the focus map. For simplicity, the input images are transformed into gray scale ones. Note that the two sequences of CBR and 8 residual blocks share weights for images A and B. As for the network configuration, there are 128 filters for each convolutional unit except the last one whose number of input and output channels are 256 and 1, respectively.

We train the network on an image segmentation dataset, PASCAL VOC 2012. The segmentation map is regarded as the ground truth focus map, and the α-matte method [22] is utilized to generate the multi-focus images. Firstly, the clear foreground ($\mathrm{FG^C}$) and background ($\mathrm{BG^C}$) regions are blurred by Gaussian filters and their blurred versions are denoted by $\mathrm{FG^B}$ and $\mathrm{BG^B}$. When the foreground is focused, the source image is simulated by the original focus map α^{C}. Otherwise, the source image is simulated by the blurred focus map α^{B}. In formula, there are

$$\begin{aligned} \boldsymbol{X}^{[1]} &= \mathrm{FG^C} + (1 - \alpha^{\mathrm{C}})\mathrm{BG^B}, \\ \boldsymbol{X}^{[2]} &= \mathrm{FG^B} + (1 - \alpha^{\mathrm{B}})\mathrm{BG^C}. \end{aligned} \tag{16}$$

There are 2913 pairs of images in total, so we utilize the image rotation to augment the data. Our network is optimized by Adam over 50 epochs with a batch size of 6 and a learning rate of 10^{-4}. The loss function is the binary cross-entropy between outputs and ground truth focus maps.

III. Experiments

We conduct extensive experiments on the real-world dataset to study behaviors and properties of the MFF-SSIM based fusion strategy. In what follows, our methods are abbreviated as MS-Lap and MS-ResNet.

A. Datasets and Metrics

Our experiments' goal is to verify whether our framework does a better job than other methods if images suffer from severe defocus spread effects. In the following experiments, we employ the MFFW dataset [24] to evaluate the algorithms' perfromance. This dataset is presented recently and it contains 13 pairs of real-world multi-focus images collected on the Internet. This dataset provides annotated focus maps for each pair. In addition, to facilitate assessment, this dataset also releases the manually fused images. The scenes in MFFW are far more complicated and there is a significant defocus spread effect. It is a challenge to obtain satisfactory fusion images on this dataset.

Normalized mutual information (NMI), Xydeas's metric [28] and Chen-Blum's metric [29] are employed as the no-reference based metrics. Since MFFW [24] provides the manually fused images, we use three reference image quality metrics to assess the algorithm performance, that is, peak signal-to-noise ratio (PSNR), SSIM [25] and feature similarity index (FSIM) [30]. Furthermore, except for these metrics we also report the mean opinion score (MOS). In detail, 10 volunteers were invited to

TABLE I
THE RESULTS ON MFFW DATASET. THE BEST AND THE SECOND BEST VALUES ARE HIGHLIGHTED BY BOLD TYPEFACE AND UNDERLINE, RESPECTIVELY

Methods	Objective Metric	Reference Based Metrics			No Reference Based Metrics		
	MOS	PSNR	SSIM	FSIM	NMI	Xydeas	Chen-Blum
BF	7.3846	34.0595	0.9833	0.9809	1.1104	0.5941	**0.7456**
BRW	7.3846	35.4858	0.9865	0.9839	1.0415	0.6165	0.7268
CBF	6.6923	32.7452	0.9741	0.9694	0.8648	0.5244	0.6554
GFF	8.4615	35.1882	0.9848	0.9851	0.9371	0.6022	0.7186
CNN	8.5385	35.1722	0.9829	0.9831	1.0638	**0.6615**	0.7362
ECNN	7.9015	35.0610	0.9845	0.9829	**1.1353**	0.6135	0.7321
DPRL	7.3723	32.8121	0.9796	0.9751	1.1214	0.6159	0.7124
MMF-Net	7.3077	31.8415	0.9764	0.9709	0.9174	0.4243	0.6644
MS-ResNet	8.6154	**36.4105**	**0.9882**	**0.9888**	0.9848	0.5858	0.7233
MS-Lap	**8.9231**	35.6802	0.9860	0.9880	1.0151	0.6082	0.7335

(a) Source 1 (b) Source 2 (c) Reference

Fig. 5. (a)–(b) No. 6 image set of the MFFW dataset. (c) The manually fused image.

evaluate the quality of fused images. All volunteers had no bias about this task. Their opinion score ranged from 1 to 10, and larger values corresponded to better images.

B. Comparison With SOTA Methods

Our technique is compared with eight SOTA methods, including boundary founding (BF) [31], BRW [32], CBF [33], GFF [5], CNN [15], DPRL [17], ECNN [18] and MMFNet [22]. The metrics are reported in Table I. The best and the second best values are highlighted by bold face and underline, respectively. It is shown that MS-ResNet achieves the highest PSNR, SSIM and FSIM values, and the second highest MOS value. MS-Lap has the best MOS value, and the second best PSNR, FSIM and Chen-Blum values. The no-reference based metrics show that our methods are comparable with popular counterparts as well. In the contrast, MMFNet almost performs worst, although the experiment in reference [22] has proofed that MFF-Net outperforms others if the image suffers from mild defocus spread effects.

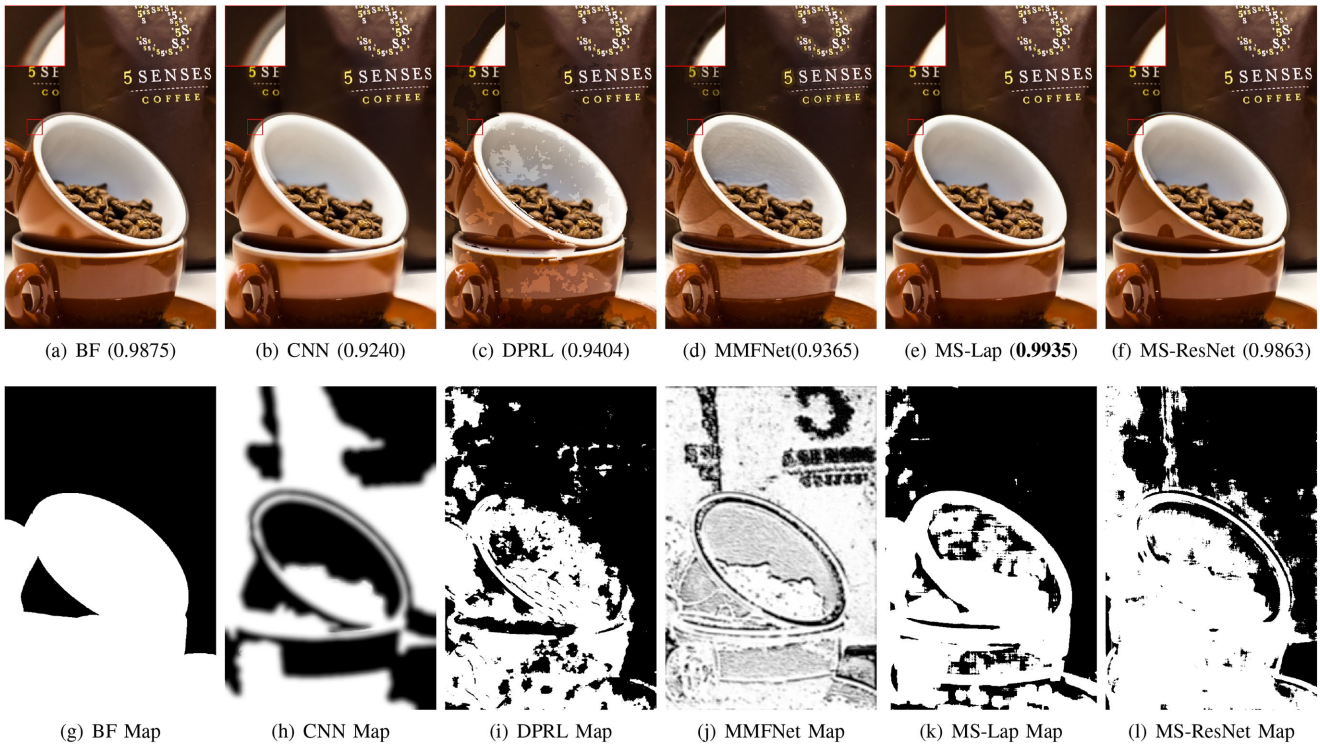
Besides the quantitative comparison, representative fusion images are visualized to further exhibit the effectiveness of MFF-SSIM based methods. The No. 6 image pair is displayed in Fig. 5. Two cups and two coffee bags are in near and distant focuses, respectively. Owing to the defocus spread effect, the characters on bags and the edges of two cups expand in source 1 and source 2 images, respectively. The fusion images and detected focus maps are displayed in Fig. 6. It is shown that DPRL and MMFNet break down because they fragment the scene into many pieces and result in the obvious artifacts and ghosts. It indicates that their performances highly depend on the detector's accuracy. The fusion images generated by BF and CNN look more pleasant. However, for CNN, only the coffee beans and bags are clear, and most of the regions in two cups are still blurred; for BF, on account of defocus spread effects there are conspicuous haloes around the cups, so it cannot match up our expectations either. Obviously, MS-Lap and MS-ResNet generate the most satisfactory images, because all objects are clear and without artifacts or ghosts. At the same time, we can see that the focus maps detected by DPRL, MS-Lap and MS-ResNet are not accuracy enough, while DPRL fails in this case. The fact demonstrates that our proposed MFF-SSIM framework contributes more rather than the map detectors (i.e., the Laplacian energy and the ResNet).

According to the quantitative comparison and visual inspection, the conclusion can be drawn that MFF-SSIM based methods outperform others when the images suffer from defocus spread effects.

C. Robustness Experiments

Here, more experiments are conducted to analysis our proposed method. Firstly, we manually annotate the focus map, which is deemed to be accurate. Then, each pixel in this map is corrupted with a probability p. Note that corruption strategy is that the focused (defocused) pixel is changed to defocused (focused) one. Obivously the map is more inaccurate with greater p. At last, the corrupted map is taken as the input of MMF-SSIM fusion strategy. Similar steps are carried out for MMFNet. Our goals in this experiment are two-fold. The one is to observe how MFF-SSIM performs as the detected focus map getting inaccurate. The other one is to investigate whether MMF-SSIM fusion framework is more effective than the boundary post-processor in MMFNet.

This experiment is conducted 100 times, and the average PSNR/SSIM curves are displayed in Fig. 7, where corruption probability p ranges from 0 to 0.5 with a step 0.1. There is no doubt that the curves have downward tendencies as p going greater, but the PSNR and SSIM values of MFF-SSIM model are always higher than those of MMFNet. In addition, we learn that when p increases from 0 to 0.1, the PSNR value of MFF-SSIM model decreases from 49.91dB to 43.53dB by 12.78%,



(a) BF (0.9875) (b) CNN (0.9240) (c) DPRL (0.9404) (d) MMFNet(0.9365) (e) MS-Lap (**0.9935**) (f) MS-ResNet (0.9863)

(g) BF Map (h) CNN Map (i) DPRL Map (j) MMFNet Map (k) MS-Lap Map (l) MS-ResNet Map

Fig. 6. (a)–(f): Fusion images. SSIM values are shown in the parentheses. (g)–(l): Detected focus maps. The reference and source images are displayed in Fig. 5. The manually annotated focus map is shown in Fig. 7(b).

while that of MMFNet dramatically decreases from 37.87dB to 25.72dB by 32.08%. The fusion images with $p = 0.3$ are also visualized in Fig. 7. Our fusion image can match up our expectations. Nonetheless, the artifacts can be easily found in the image fused by MMFNet. Based on the above analysis, we can draw the conclusion that MFF-SSIM model is more robust than MMFNet.

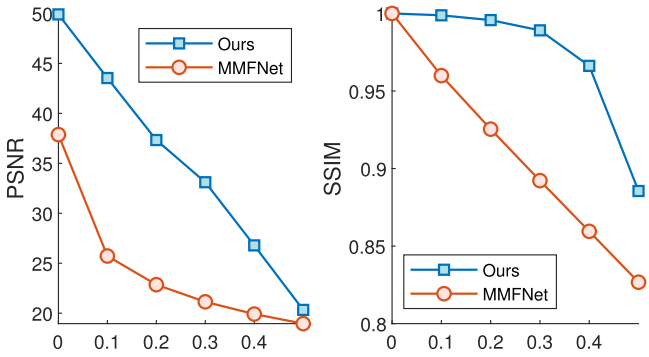
D. Ablation Experiments

In this subsection, a series of ablation experiments are conducted, that is, changing some hyper-parameters of the model and seeing how it affects performance.

1) Network Depth and Width: As shown in Fig. 4, MS-ResNet employs 2 convolutional units and 8 residual blocks, and there are 128 filters except for the last convolutional unit. Here we analyze effects of network depth (that is, the number of residual blocks) and width (that is, the number of filters). The top panel of Fig. 8 displays the PSNR and SSIM curves on the MFFW dataset with the number of residual blocks increasing from 3 to 10. It is shown that both PSNR and SSIM go larger and then tend to be flat with blocks' number growing. Eight blocks correspond to the best results. The bottom panel of Fig. 8 shows the PSNR and SSIM curves with different numbers of filters, including 8, 16, 32, 64, 128 and 256. A similar conclusion can be drawn that both PSNR and SSIM go greater with filters' number growing. Nonetheless, it is found that larger blocks' or filters' number does not necessarily improve the performance of MS-ResNet. The reason may be that the MS-ResNet suffers from the overfitting problem when the network depth or width is too large.

2) Window Size: In above experiments, the window size is empirically set as $W = \alpha MN$, where the window size ratio is set as $\alpha = 5 \times 10^{-5}$, and M and N denote the height and width of the image, respectively. Generally speaking, the configuration of window size is important to patch-wise methods. In this experiment, it aims at investigating the performance of MFF-SSIM-Strategy with different window size ratios. For simplicity, the LAP is employed as the focus map detector, and two pairs of images from MFFW are taken as representative examples.

The window size ratio is sampled from 1.5×10^{-5} to 9.5×10^{-5} with step 1×10^{-5}. Figs. 9 and 10 display the PSNR and SSIM curves, and the fusion images with the smallest, largest and optimal window sizes. In the first example, the defocus spread effect is relatively mild. It is found that as α increasing both PSNR and SSIM get larger and then decrease. The best result corresponds to $\alpha = 3.5 \times 10^{-5}$. As for visual inspection, Fig. 9(d) reveals that small window size suffers from artifacts. In addition, it is observed that the fusion image with large window size (see Fig. 9(f)) is visually similar to optimal fusion image (see Fig. 9(e)). In the second example, the defocus spread effect is relatively severe. From Fig. 10, it is learned that both PSNR and SSIM increase as window size increasing, and the best result is reached at $\alpha = 9.5 \times 10^{-5}$. And the fusion image with larger window size is visually better than that with smaller window size. In summary, larger the window size is, better our method is. This conclusion matches up our anticipation,



(a) Metric curves (versus corruption probability p)

(b) Annotated Map (c) Corrupted Map (d) Ours

(e) Annotated Matte Map (f) Corrupted Matte Map (g) MMFNet

Fig. 7. Analysis on the focus map. (a) The metrics with different corruption probabilities. (b)–(d) The results of our method when $p = 0.3$. (e)–(g) The results of MMF-Net when $p = 0.3$.

to some degree, because larger window size indicates that there are more neighbors help the center point to determine its pixel value.

However, this conclusion will not stay true if the foreground or background is disconnected. For example, as shown in Fig. 11 larger window size would do harm to the fusion image, when there is a crossed fence. In addition, it is worthy pointing out that larger window size significantly raises execution time. It still remains a problem that how to automatically pick optimal window size so as to strike the balance between performance and running speed.

E. Multiple Source Images

The above experiments mainly focus on the case of two source images. It is interesting to study the performance of our method if there are multiple images (i.e., $K > 2$). Recall the updating rule, viz. Eq. (14). We know that if the focus map detector can

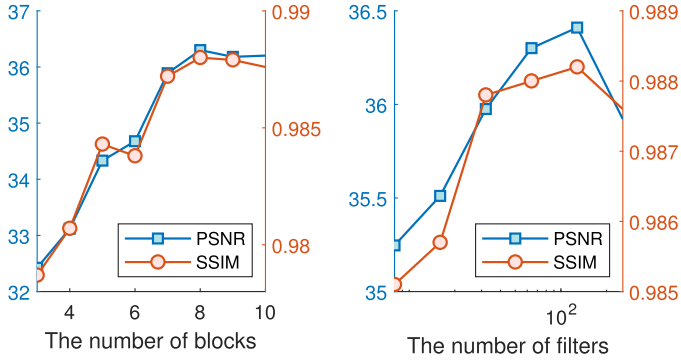
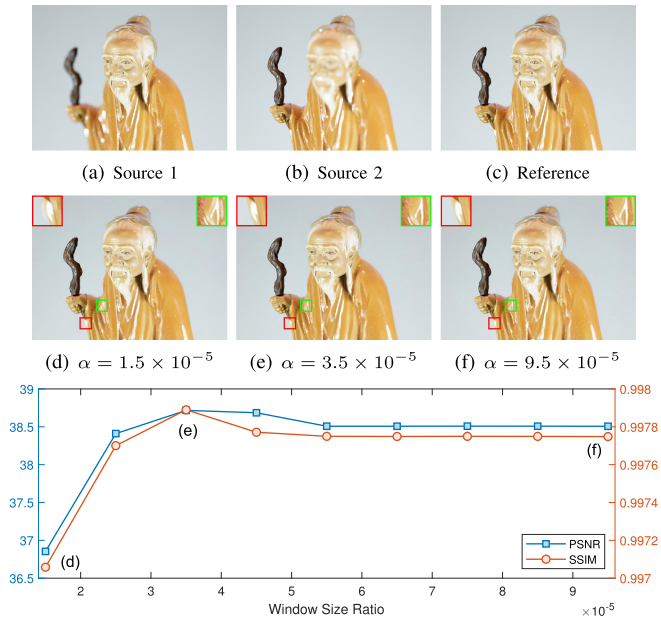
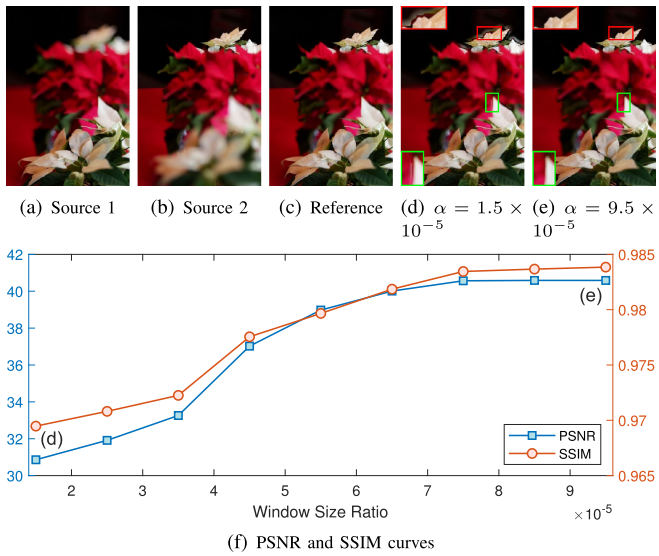


Fig. 8. Effects of the network depth (left) and width (right).



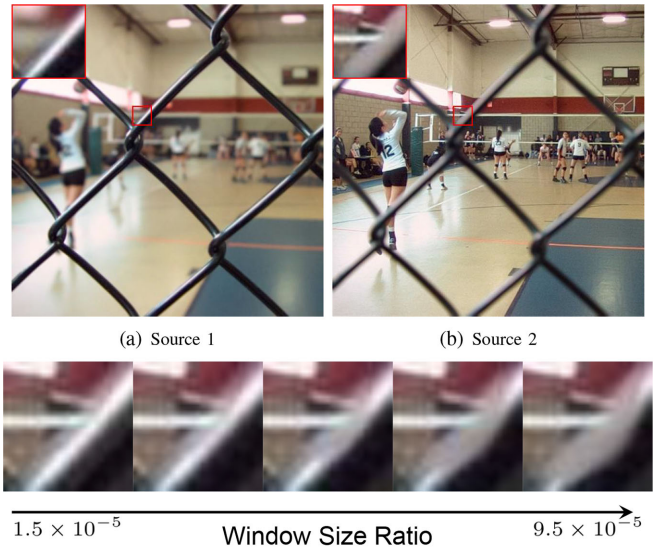
(g) PSNR and SSIM curves

Fig. 9. The results with different window size ratios of No. 4 image set from MFFW.



(f) PSNR and SSIM curves

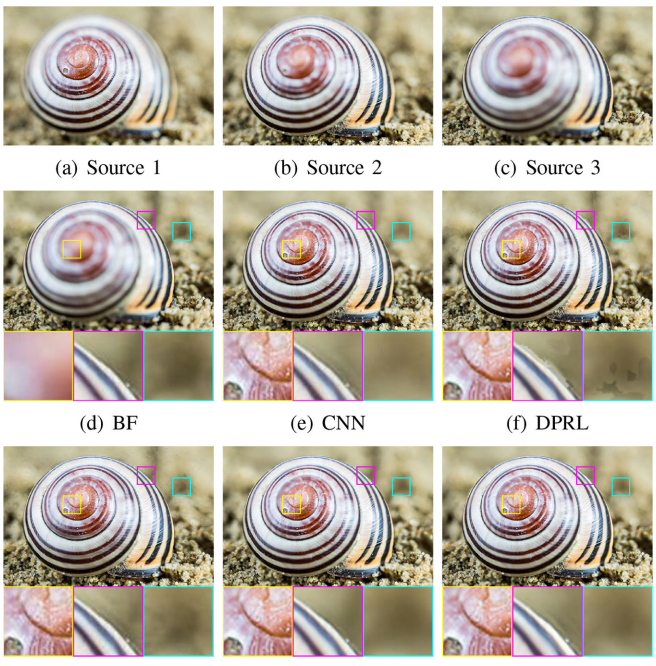
Fig. 10. The results with different window size ratios of No. 11 image set from MFFW.



(a) Source 1 (b) Source 2

(c) Local patches

Fig. 11. The results with different window size ratios of No. 5 image set from Lytro. Since there is no ground truth, we only display the concerned local patches instead of the PSNR and SSIM curves.



(a) Source 1 (b) Source 2 (c) Source 3

(d) BF (e) CNN (f) DPRL

(g) MMFNet (h) MS-Lap (i) MS-ResNet

Fig. 12. Fusion results of a set of multiple source images.

generate the maps for K source images simultaneously (e.g. the Laplace energy method), MFF-SSIM model will be directly applied in this case. However, if the focus map detector only deals with two source images (e.g. ResNet), we have to fuse them one by one. A representative image set is displayed in Fig. 12. There is a shell, whose different parts are in the near, middle and distant focuses. It is shown that our methods still provide satisfactory images.

F. Mild Defocus Spread Effect Experiments

Though the mild defocus spread effect is out of our scope, it is also interesting to investigate the behavior of our methods in this case. To this end, the algorithms are applied to the Lytro and Grayscale datasets who suffer from mild defocus spread effects. With the limitation of paper length, the results are displayed in supplementary materials. Although our methods are not the best performer on Lytro and Grayscale datasets, our fusion images are artifact-free and are visually similar to the best one (i.e., MMFNet).

IV. Conclusion

This paper presents an SSIM-based multi-focus image fusion framework, the first attempts to deal with severe defocus spread effects. The experimental results show that our framework outperforms the SOTA methods. Our fusion images are artifact-free, while others contains obvious artifacts. However, our method is time-consuming. It is interesting to investigate how to design a real-time algorithm to deal with defocus spread effects in the future. And, light filed imaging provides the richer visual information than the classic photography and it has been applied to depth estimation and super-resolution [34], [35]. Therefore, another interesting future work is how to effectively combine multi-focus images with the light field data to overcome the defocus spread effects.

References

[1] Q. Zhang, Y. Liu, R. S. Blum, J. Han, and D. Tao, "Sparse representation based multi-sensor image fusion for multi-focus and multi-modality images: A review," *Inf. Fusion*, vol. 40, pp. 57–75, 2018.
[2] Y. Li and S. Jiang, "Multi-focus image fusion using geometric algebra based discrete fourier transform," *IEEE Access*, vol. 8, pp. 60 019–60 028, 2020.
[3] G. Pajares and J. M. de la Cruz, "A wavelet-based image fusion tutorial," *Pattern Recognit.*, vol. 37, no. 9, pp. 1855–1872, Sep. 2004.
[4] Q. Zhang and B. long Guo, "Multifocus image fusion using the nonsubsampled contourlet transform," *Signal Process.*, vol. 89, no. 7, pp. 1334–1346, 2009.
[5] S. Li, X. Kang, and J. Hu, "Image fusion with guided filtering," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2864–2875, Jul. 2013.
[6] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[7] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.
[8] B. Yang and S. Li, "Multifocus image fusion and restoration with sparse representation," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 4, pp. 884–892, Apr. 2010.
[9] Y. Liu and Z. Wang, "Simultaneous image fusion and denoising with adaptive sparse representation," *IET Image Process.*, vol. 9, no. 5, pp. 347–357, 2015.
[10] Y. Liu, X. Chen, R. K. Ward, and Z. Jane Wang, "Image fusion with convolutional sparse representation," *IEEE Signal Process. Lett.*, vol. 23, no. 12, pp. 1882–1886, Dec. 2016.
[11] Y. Liu, S. Liu, and Z. Wang, "A general framework for image fusion based on multi-scale transform and sparse representation," *Inf. Fusion*, vol. 24, pp. 147–164, 2015.
[12] X. Qiu, M. Li, L. Zhang, and X. Yuan, "Guided filter-based multi-focus image fusion through focus region detection," *Signal Process. Image Commun.*, vol. 72, pp. 35–46, 2019.
[13] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[14] S. Li, X. Kang, J. Hu, and B. Yang, "Image matting for fusion of multi-focus images in dynamic scenes," *Inf. Fusion*, vol. 14, no. 2, pp. 147–162, 2013.

[15] Y. Liu, X. Chen, H. Peng, and Z. Wang, "Multi-focus image fusion with a deep convolutional neural network," *Inf. Fusion*, vol. 36, pp. 191–207, 2017.
[16] Y. Zhang, Y. Liu, P. Sun, H. Yan, X. Zhao, and L. Zhang, "Ifcnn: A general image fusion framework based on convolutional neural network," *Inf. Fusion*, vol. 54, pp. 99–118, 2020.
[17] J. Li *et al.*, "DRPL: Deep regression pair learning for multi-focus image fusion," *IEEE Trans. Image Process.*, vol. 29, pp. 4816–4831, 2020.
[18] M. Amin-Naji, A. Aghagolzadeh, and M. Ezoji, "Ensemble of cnn for multi-focus image fusion," *Inf. Fusion*, vol. 51, pp. 201–214, 2019.
[19] Z. Nian and C. Jung, "Cnn-based multi-focus image fusion with light field data," in *Proc. Int. Conf. Image Process.*, Taipei, Taiwan, Sep. 2019, pp. 1044–1048.
[20] Y. Liu, L. Wang, J. Cheng, C. Li, and X. Chen, "Multi-focus image fusion: A survey of the state of the art," *Inf. Fusion*, vol. 64, pp. 71–91, 2020.
[21] X. Zhang, "Multi-focus image fusion: A benchmark," 2020, *arXiv:2005.01116*.
[22] H. Ma, Q. Liao, J. Zhang, S. Liu, and J. H. Xue, "An α-matte boundary defocus model-based cascaded network for multi-focus image fusion," *IEEE Trans. Image Process.*, vol. 29, pp. 8668–8679, 2020.
[23] H. Ma, J. Zhang, S. Liu, and Q. Liao, "Boundary aware multi-focus image fusion using deep neural network," in *Proc. IEEE Int. Conf. Multimedia Expo*, Shanghai, China, Jul. 2019, pp. 1150–1155.
[24] S. Xu, X. Wei, C. Zhang, J. Liu, and J. Zhang, "MFFW: A new dataset for multi-focus image fusion," 2020, *arXiv:2005.01116*.
[25] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[26] K. Ma, K. Zeng, and Z. Wang, "Perceptual quality assessment for multi-exposure image fusion," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3345–3356, Nov. 2015.
[27] K. Ma, Z. Duanmu, H. Yeganeh, and Z. Wang, "Multi-exposure image fusion by optimizing a structural similarity index," *IEEE Trans. Comput. Imag.*, vol. 4, no. 1, pp. 60–72, Mar. 2018.
[28] C. S. Xydeas and V. Petrovic, "Objective image fusion performance measure," *Electron. Lett.*, vol. 36, no. 4, pp. 308–309, 2000.
[29] Y. Chen and R. S. Blum, "A new automated quality assessment algorithm for image fusion," *Image Vis. Comput.*, vol. 27, no. 10, pp. 1421–1432, 2009, special Section: Computer vision methods for ambient intelligence.
[30] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
[31] Y. Zhang, X. Bai, and T. Wang, "Boundary finding based multi-focus image fusion through multi-scale morphological focus-measure," *Inf. Fusion*, vol. 35, pp. 81–101, 2017.
[32] J. Ma, Z. Zhou, B. Wang, L. Miao, and H. Zong, "Multi-focus image fusion using boosted random walks-based algorithm with two-scale focus maps," *Neurocomputing*, vol. 335, pp. 9–20, 2019.
[33] B. K. Shreyamsha Kumar, "Image fusion based on pixel significance using cross bilateral filter," *Signal Image Video Process.*, vol. 9, no. 5, pp. 1193–1204, Jul. 2015.
[34] H. Sheng, S. Zhang, X. Cao, Y. Fang, and Z. Xiong, "Geometric occlusion analysis in depth estimation using integral guided filter for light-field image," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5758–5771, 2017.
[35] G. Wu, Y. Liu, Q. Dai, and T. Chai, "Learning sheared EPI structure for light field reconstruction," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3261–3273, Jul. 2019.