Deep Guided Learning for Fast Multi-Exposure Image Fusion

Kede Ma, *Member, IEEE,* Zhengfang Duanmu, *Student Member, IEEE,* Hanwei Zhu, *Student Member, IEEE,* Yuming Fang, *Senior Member, IEEE,* and Zhou Wang, *Fellow, IEEE*

***Abstract*—We propose a fast multi-exposure image fusion (MEF) method, namely MEF-Net, for static image sequences of arbitrary spatial resolution and exposure number. We first feed a low-resolution version of the input sequence to a fully convolutional network for weight map prediction. We then jointly upsample the weight maps using a guided filter. The final image is computed by a weighted fusion. Unlike conventional MEF methods, MEF-Net is trained end-to-end by optimizing the perceptually calibrated MEF structural similarity (MEF-SSIM) index over a database of training sequences at full resolution. Across an independent set of test sequences, we find that the optimized MEF-Net achieves consistent improvement in visual quality for most sequences, and runs 10 to 1000 times faster than state-of-the-art methods. The code is made publicly available at https://github.com/makedede/MEFNet.**

***Index Terms*—Multi-exposure image fusion, convolutional neural networks, guided filtering, computational photography.**

I. Introduction

MULTI-EXPOSURE image fusion (MEF) provides a cost-effective solution for high-dynamic-range (HDR) imaging [1]. It takes an image sequence with different exposure levels as input and produces a high-quality and low-dynamic-range image, ready for display [2]. Research in MEF has yielded a number of methods [2]–[8], which generate fused images with faithful detail preservation and vivid color appearance. This is mainly accomplished by a weighted summation framework

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{W}_k \odot \mathbf{X}_k\,, \tag{1}$$

where $\odot$ denotes the Hadamard product. $\mathbf{W}_k$ and $\mathbf{X}_k$ represent the k-th weight map and the corresponding exposure image, respectively, $\mathbf{Y}$ is the fused image, and K is the number of exposures in the input sequence. Noticeable exceptions of the framework are optimization-based methods [6], [8], where the fusion process is supervised by a perceptual image quality metric [9].

Despite the demonstrated success, the high resolution of the exposure sequence captured by commercial cameras and mobile devices poses a grand challenge to existing MEF methods, which may require extensive computational resources and take seconds (or even minutes) to generate the fused results. The situation becomes even worse with the increasing number of exposures. Algorithm acceleration through code optimization is possible [10], [11], but it may not generalize across different MEF methods. Another general approach to accelerate an MEF method [12]–[14] is to downsample the input sequence, execute the MEF operator at low resolution, and upsample the fused image. One drawback of this approach is that the MEF method never sees the high-resolution sequences and therefore fails to fully reproduce the fine details, limiting the visual sharpness of the fused images.

We aim to develop an MEF method for static scenes with three desirable properties:

- *Flexibility*. It must accept input sequences of arbitrary spatial resolution and exposure number.
- *Speed*. It must be fast, facilitating real-time mobile applications at high resolution.
- *Quality*. It must produce high-quality fused images across a broad range of content and luminance variations.

To achieve *flexibility*, we utilize a fully convolutional network [15], which takes an input of arbitrary size and produces an output of the corresponding size (known as dense prediction). The network is shared by different exposed images, enabling it to process an arbitrary number of exposures. To achieve *speed*, we follow the downsample-execute-upsample scheme and feed the network a low-resolution version of the input sequence. Rather than producing the fused image as in [6], [16], [17], the network learns to generate the low-resolution weight maps in Eq. (1) and jointly upsample them using a guided filter [18] for final weighted fusion. By doing so, we take advantage of the smooth nature of the weight maps and make use of the input sequence as the guidance [19]. Directly upsampling the fused image is difficult due to the existence of rich high-frequency information in the high-resolution sequence and the lack of proper guidance. To achieve *quality*, we integrate the differentiable guided filter with the preceding network [19] and optimize the entire model end-to-end for the subject-calibrated MEF structural similarity (MEF-SSIM) index [9] over a large number of training sequences [6], [20]–[23]. Although most of our inference and learning is performed at low resolution, the objective function MEF-SSIM [9] is measured at full resolution, which encourages the guided filter to cooperate with the convolutional network, generating high-quality fused images. Extensive experiments demonstrate that the resulting MEF-Net achieves consistent improvement in

Kede Ma is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: kede.ma@cityu.edu.hk).

Zhengfang Duanmu and Zhou Wang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: zduanmu@uwaterloo.ca; zhou.wang@uwaterloo.ca).

Hanwei Zhu and Yuming Fang are with the School of Information Management, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi, China (e-mail: hanwei.zhu@outlook.com; fa0001ng@e.ntu.edu.sg).

visual quality compared with state-of-the-art MEF methods for most sequences. More importantly, MEF-Net runs 10 to 1000 times faster and holds much promise for approximating and accelerating the MEF methods that are computationally intensive.

II. Related Work

In this section, we provide a brief overview of existing MEF methods and general approaches for fast image processing, with emphasis on previous ones that are closely related to our work.

A. Existing MEF Algorithms

The Laplacian pyramid [24] proposed by Burt and Adelson in 1983 has a lasting impact on image fusion research [25]. Combining with Gaussian [2], [3] or edge-preserving [4], [7] filters, the Laplacian pyramid provides a convenient multi-resolution framework to refine the weight map $\mathbf{W}_k$, which carries perceptually important information of $\mathbf{X}_k$. Mertens *et al.* [2] adopted this framework and proposed one of the first pixel-wise MEF methods, which keeps a good balance between visual quality and computational complexity. Since then, a great number of pixel-wise MEF methods [26] have been developed, mainly to improve visual quality at the cost of increasing computational complexity. Compared to pixel-wise MEF, patch-wise methods generally produce a smoother $\mathbf{W}_k$ that requires less post-processing, but bear heavier computational burdens. Goshtasby [27] presented one of the first patch-wise MEF methods. Ma and Wang [28] extended the idea [27] and developed a structural patch decomposition for MEF. Typical perceptual factors that contribute to $\mathbf{W}_k$ include gradient [29], contrast [2], color saturation [2], [7], entropy [27], structure [28], well-exposedness [2], [3], and saliency [4].

B. Fast Image Processing

As mobile devices become people's primary cameras to take photos, there is a growing demand to accelerate image processing operators for novel mobile applications such as photo editing, face manipulation, and augmented reality. A good case in point is bilateral filtering [30]–[32], which benefits from years of code optimization, due to the ubiquity of edge-preserving image processing. However, such acceleration tricks may not generalize to other operators. A system-level acceleration solution, friendly to mobile hardware, is to send images to a cloud server, execute the image processing operator on the cloud, and send the processed images back [33]. Due to the large bitrate of high-resolution images, this may introduce significant delays, especially when the network condition is unstable. The downsample-execute-upsample scheme is another general method for algorithm acceleration, which suffers from two limitations. First, the underlying operator may still be slow to run at low resolution. Second, it is difficult for upsampling techniques to recover the high-frequency information in the high-resolution images, especially when they are of complex structures. Recently, due to efficient feed-forward inference, convolutional networks [14], [19] have been used to approximate and accelerate popular image processing operators, including edge-preserving filtering, detail manipulation, non-local dehazing, and style transfer.

C. Closely Related Work

Our work is closely related to several previous methods. Li *et al.* [4] first introduced guided filtering to MEF. The weight map $\mathbf{W}_k$ was constructed based on pixel saliency and spatial consistency measurements, and was refined by a guided filter. Kou *et al.* [7] built their work upon [2] and replaced Gaussian smoothing with gradient domain guided filtering. The three components of the above two methods—weight map construction, guided filtering, and weighted fusion—are optimized separately (often through manual adjustment). Our method is different from them by resorting to an end-to-end solution, where the three components are jointly optimized in a data-driven fashion. Rather than pre-defining a computational graph for MEF, Ma *et al.* [8] formulated it as an optimization problem

$$\begin{aligned} \mathbf{Y}_{\text{opt}} = \arg\max_{\mathbf{Y}} \quad & \text{MEF-SSIM}(\{\mathbf{X}_k\}, \mathbf{Y}) \\ \text{subject to} \quad & 0 \le \mathbf{Y} \le 255\,. \end{aligned} \tag{2}$$

Due to the nonconvexity of MEF-SSIM [9] and the high-dimensionality of the optimization problem, a closed-form solution is difficult. Therefore, a gradient-based iterative solver is adopted [8], which is computationally expensive. Another work closely related to ours is from Prabhakar *et al.* [6], who trained a feed-forward convolutional network to solve the optimization problem in (2). The method works reasonably well on extreme situations, but does not achieve the *flexibility*, *speed*, and *quality* we seek. We will show that the proposed MEF-Net achieves higher quality, while being much faster and more flexible.

Chen *et al.* [14] investigated a number of convolutional network architectures in terms of their approximation accuracy, speed, and compactness when accelerating image processing operators. They found that a multi-scale context aggregation network (CAN) characterized by dilated convolutions [34] satisfies the three criteria and significantly outperforms prior methods [13]. We will adopt CAN as our default network architecture. Wu *et al.* [19] treated the guided filter as a group of spatially-varying differentiable transformations and integrated it with convolutional networks for end-to-end training. Although their method [19] achieves superior performance in some applications with relatively smooth outputs (*e.g.*, style transfer [35]), it cannot accurately approximate operators that work with high-frequency image content (*e.g.*, multi-scale tone manipulation [36]). Our method circumvents this problem by applying the guided filter on $\mathbf{W}_k$, which is smoother and easier to upsample than $\mathbf{Y}$.

III. MEF-Net

We describe MEF-Net, a flexible, fast, and high-quality MEF method. MEF-Net consists of a bilinear downsampler, a CAN, a guided filter, and a weighted fusion module. The

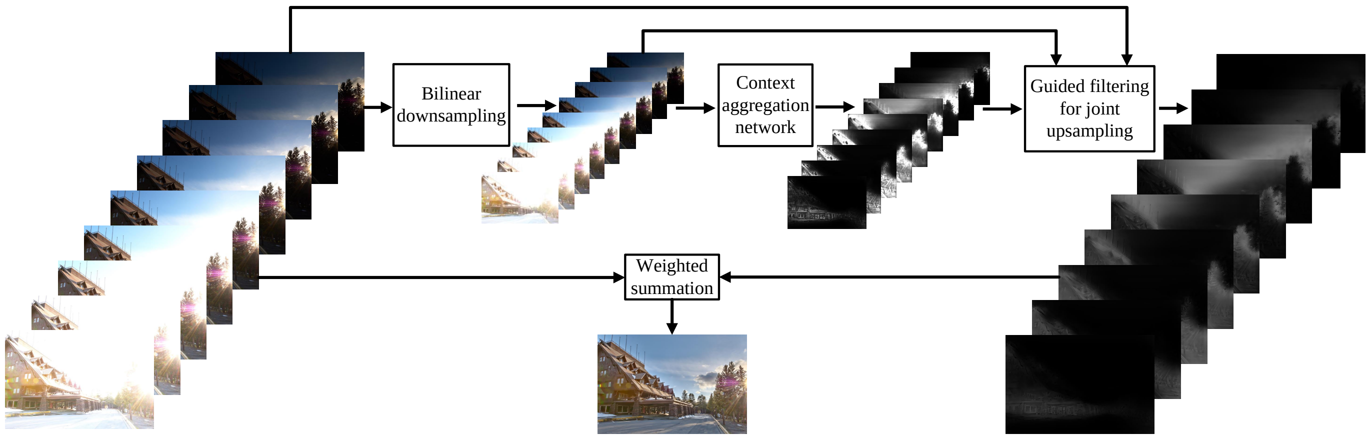

Bilinear downsampling
Context aggregation network
Guided filtering for joint upsampling
Weighted summation

Fig. 1. Schematic diagram of the proposed MEF-Net. The downsampled input sequence $\{\mathbf{X}_k^l\}$ is fed to CAN, where the main computation takes place. The learned weight maps $\{\mathbf{W}_k^l\}$ are jointly upsampled to high resolution by the guided filter. The fused image is obtained by a weighted summation of $\{\mathbf{X}_k\}$ and $\{\mathbf{W}_k\}$.

TABLE I
SPECIFICATION OF CAN IN MEF-NET FOR LOW-RESOLUTION WEIGHT MAP PREDICTION

Layer	1	2	3	4	5	6	7
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	2	4	8	16	1	1
Width	24	24	24	24	24	24	1
Bias	✗	✗	✗	✗	✗	✗	✓
Adaptive normalization	✓	✓	✓	✓	✓	✓	✗
Nonlinearity	✓	✓	✓	✓	✓	✓	✗
Receptive field	3×3	7×7	15×15	31×31	63×63	65×65	65×65

architecture is shown in Fig. 1. We first downsample an input sequence $\{\mathbf{X}_k\}$ and feed CAN the low-resolution version $\{\mathbf{X}_k^l\}$ to predict the low-resolution weight maps $\{\mathbf{W}_k^l\}$. Taking $\{\mathbf{W}_k^l\}$, $\{\mathbf{X}_k^l\}$, and $\{\mathbf{X}_k\}$ as inputs, we obtain the high-resolution weight maps $\{\mathbf{W}_k\}$ using the guided filter, an operation also known as joint upsampling in computer vision [37]. Finally, we compute the fused image $\mathbf{Y}$ using Eq. (1). MEF-Net is end-to-end trainable with the objective function MEF-SSIM [9] evaluated at high resolution.

A. CAN for Low-Resolution Weight Map Prediction

The core module of MEF-Net is a convolutional network, which transforms the low-resolution input sequence $\{\mathbf{X}_k^l\}$ to the corresponding weight maps $\{\mathbf{W}_k^l\}$. The network must allow for $\{\mathbf{X}_k^l\}$ of arbitrary spatial size and exposure number, and produce $\{\mathbf{W}_k^l\}$ of the corresponding size and number. To achieve this, we make use of a fully convolutional network to handle all exposures (*i.e.*, images of different exposures share the same weight generation network), which can be efficiently implemented by allocating $\{\mathbf{X}_k^l\}$ along the batch dimension. From a number of alternative networks [15], [38], we select CAN [34], which has been advocated by Chen *et al.* [14] and Wu *et al.* [19] for approximating image processing operators. The key advantage of CAN is its large receptive field without sacrificing spatial resolution. It gradually aggregates contextual information at deeper layers and accomplishes computation of global image statistics for better image modeling. Table I specifies our CAN configuration. It has seven convolution layers, whose responses have the same resolution as the input. Similar to [14], we employ adaptive normalization right after convolution

$$\mathrm{AN}(\mathbf{Z}) = \lambda_n \mathbf{Z} + \lambda_n' \mathrm{IN}(\mathbf{Z})\,, \tag{3}$$

where $\lambda_n, \lambda_n' \in \mathbb{R}$ are learnable scalar weights, $\mathbf{Z}$ indicates the intermediate representations, and $\mathrm{IN}(\cdot)$ stands for the instance normalization operator [39]. We choose not to use batch normalization [40] here because the batch size (*i.e.*, the number of exposures) is usually small, which may introduce problems during training due to inaccurate batch statistics estimation. In addition, to better preserve the local structural information of $\mathbf{X}_k^l$ [20], we adopt the leaky rectified linear unit (LReLU) as the point-wise nonlinearity

$$\mathrm{LReLU}(\mathbf{Z}) = \max(\lambda_r \mathbf{Z}, \mathbf{Z})\,, \tag{4}$$

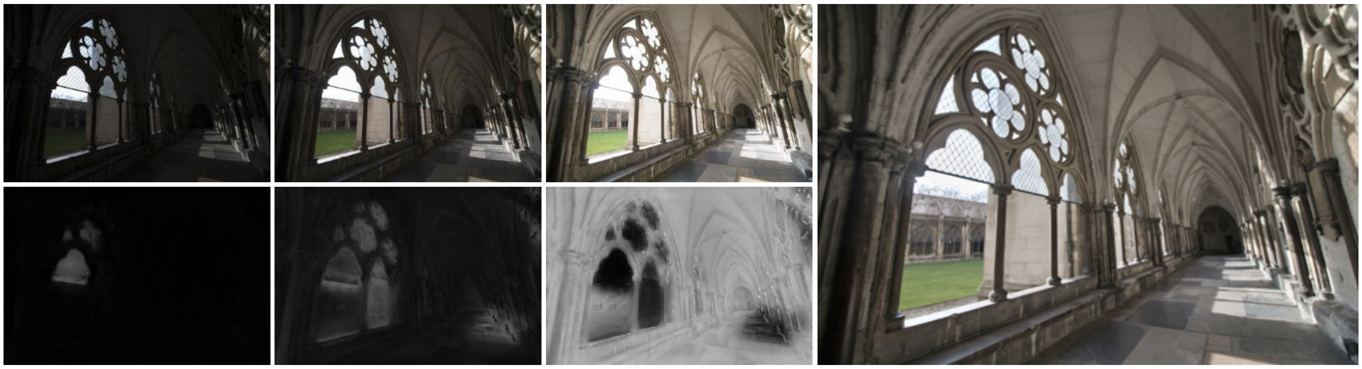
where $\lambda_r > 0$ is a fixed parameter during training. The output layer produces $\mathbf{W}_k^l$ using a 1×1 convolution without adaptive normalization and nonlinearity.

B. Guided Filter for High-Resolution Weight Map Upsampling

The key assumption of the guided filter is a local linear model between the guidance $\mathbf{I}$ and the filtering output $\mathbf{Q}$ [18]

$$\mathbf{Q}(i) = a_\omega \mathbf{I}(i) + b_\omega, \forall i \in \omega\,, \tag{5}$$

where i is the index of the guidance and ω is a local square window with radius r. a_ω, b_ω are linear coefficients assumed



(a)

(b)

Fig. 2. Demonstration of the learned weight map $\hat{\mathbf{W}}_k$. A brighter pixel in $\hat{\mathbf{W}}_k$ indicates that the corresponding pixel in $\mathbf{X}_k$ contributes more to the fused image $\mathbf{Y}$. $\hat{\mathbf{W}}_k$ shows a strong preference to high-contrast and well-exposed regions. (a) Source sequence "Corridor" along with the learned weight maps. (b) Fused image by MEF-Net. Sequence courtesy of Jianrui Cai.

Algorithm 1 Guided filtering for joint upsampling in MEF-Net

```
Input: Low-resolution weight map $\mathbf{W}_k^l$, low-resolution $\mathbf{X}_k^l$ as guidance, high-resolution $\mathbf{X}_k$ as guidance, radius $r$, and regularization parameter $\lambda_a$
Output: High-resolution weight map $\mathbf{W}_k$
  $\text{mean}_g = f_{\text{mean}}\left(\mathbf{X}_k^l\right)$
  $\text{mean}_i = f_{\text{mean}}\left(\mathbf{W}_k^l\right)$
  $\text{corr}_g = f_{\text{mean}}\left(\mathbf{X}_k^l \odot \mathbf{X}_k^l\right)$
  $\text{corr}_{gi} = f_{\text{mean}}\left(\mathbf{X}_k^l \odot \mathbf{W}_k^l\right)$
  $\text{var}_g = \text{corr}_g - \text{mean}_g \odot \text{mean}_g$
  $\text{cov}_{gi} = \text{corr}_{gi} - \text{mean}_g \odot \text{mean}_i$
  $\mathbf{A}_k^l = \text{cov}_{gi} \oslash (\text{var}_g + \lambda_a)$
  $\mathbf{B}_k^l = \text{mean}_i - \mathbf{A}_k^l \odot \text{mean}_g$
  $\mathbf{A}_k = f_{\uparrow}\left(\mathbf{A}_k^l\right)$
  $\mathbf{B}_k = f_{\uparrow}\left(\mathbf{B}_k^l\right)$
  $\mathbf{W}_k = \mathbf{A}_k \odot \mathbf{X}_k + \mathbf{B}_k$
```

to be constant in ω and can be computed by minimizing the reconstruction error

$$\ell(a_\omega, b_\omega) = \sum_{i \in \omega} \left((a_\omega \mathbf{I}(i) + b_\omega - \mathbf{P}(i))^2 + \lambda_a a_\omega^2 \right) , \tag{6}$$

where $\mathbf{P}$ is the filtering input and λ_a is a regularization parameter penalizing large a_ω [18]. In the context of MEF-Net, we treat the low-resolution $\mathbf{W}_k^l$ and $\mathbf{X}_k^l$ as the input and the guidance of the guided filter to obtain $\mathbf{A}_k^l$ and $\mathbf{B}_k^l$, respectively. As in [18], [19], we replace the mean filter on $\mathbf{A}_k^l$ and $\mathbf{B}_k^l$, and bilinearly upsample them to the high-resolution $\mathbf{A}_k$ and $\mathbf{B}_k$. The high-resolution weight map $\mathbf{W}_k$ is computed by

$$\mathbf{W}_k = \mathbf{A}_k \odot \mathbf{X}_k + \mathbf{B}_k \, . \tag{7}$$

Algorithm 1 summarizes the guided filter for joint upsampling in MEF-Net, where f_{mean} and $f_{\uparrow}$ denote box filtering and bilinear upsampling, respectively. By interpreting the guided filter as a group of spatially-varying differentiable transformations [19], we integrate it with the preceding CAN and optimize MEF-Net end-to-end at full resolution. We may apply the guided filter as a post-processing step without any training, but it hurts the fusion performance as will be clear in Section IV-B.

To stabilize gradients during training and to obtain consistent results, we take the absolute values of $\{\mathbf{W}_k\}$ followed by normalization such that they sum to one across exposures at each spatial location

$$\hat{\mathbf{W}}_k(i) = \frac{|\mathbf{W}_k(i)|}{\sum_{k=1}^{K} |\mathbf{W}_k(i)|} \, . \tag{8}$$

Fig. 2 demonstrates the learned weight maps $\{\hat{\mathbf{W}}_k\}$ of the source sequence "Corridor", where a brighter pixel in $\hat{\mathbf{W}}_k$ indicates that the corresponding pixel in $\mathbf{X}_k$ contributes more to the fused image $\mathbf{Y}$. The learned $\hat{\mathbf{W}}_k$ enjoys several desirable properties. First, $\hat{\mathbf{W}}_k$ is smooth with gentle transitions from sharp to flat regions. Second, $\hat{\mathbf{W}}_k$ prefers high-contrast and well-exposed regions, both of which significantly impact the perceptual quality of $\mathbf{Y}$. Third, $\hat{\mathbf{W}}_k$ reflects the global structure of $\mathbf{X}_k$ and is beneficial for large-scale detail preservation. As a result, the fused image appears natural without loss of details and presence of artifacts.

C. MEF-SSIM as Objective Function

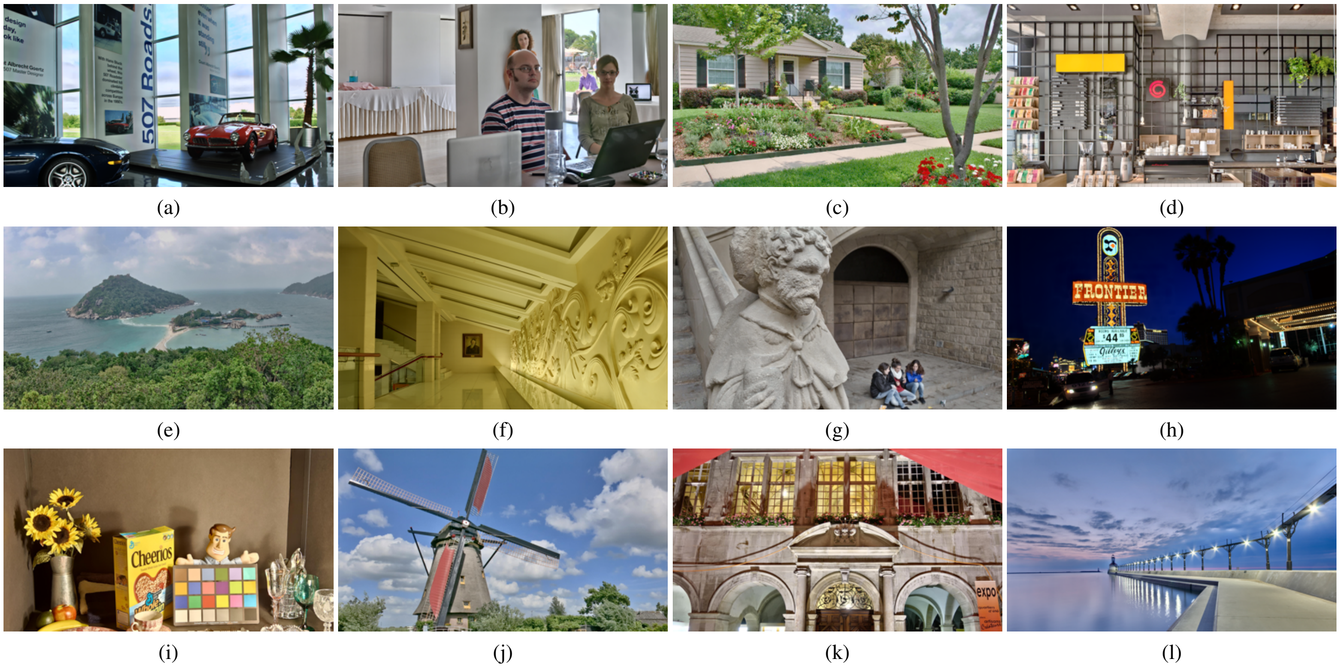
In this subsection, we detail the MEF-SSIM index [9] as the objective function for MEF-Net. Other perceptual quality metrics for MEF such as [41], [42] may also serve the purpose. Specifically, MEF-SSIM decomposes an image patch $\mathbf{x}_k$ into three conceptually independent components

$$\begin{aligned} \mathbf{x}_k &= \|\mathbf{x}_k - \mu_{\mathbf{x}_k}\| \cdot \frac{\mathbf{x}_k - \mu_{\mathbf{x}_k}}{\|\mathbf{x}_k - \mu_{\mathbf{x}_k}\|} + \mu_{\mathbf{x}_k} \\ &= \|\tilde{\mathbf{x}}_k\| \cdot \frac{\tilde{\mathbf{x}}_k}{\|\tilde{\mathbf{x}}_k\|} + \mu_{\mathbf{x}_k} \\ &= c_k \cdot \mathbf{s}_k + l_k \, , \end{aligned} \tag{9}$$

where $\|\cdot\|$ denotes the ℓ_2-norm. $l_k = \mu_{\mathbf{x}_k}$, $c_k = \|\tilde{\mathbf{x}}_k\|$, and $\mathbf{s}_k = \tilde{\mathbf{x}}_k / \|\tilde{\mathbf{x}}_k\|$ represent the intensity, contrast, and structure of $\mathbf{x}_k$, respectively [9].

The desired intensity of the fused image patch is defined by

$$\hat{l} = \frac{\sum_{k=1}^{K} w_l\left(\mu_k, l_k\right) l_k}{\sum_{k=1}^{K} w_l\left(\mu_k, l_k\right)} \, , \tag{10}$$



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 3. Sample sequences (a)–(l) gathered from five sources [6], [20]–[23]. Each sequence is represented by the corresponding fused image from MEF-Net. Images are cropped for better visibility.

where $w_l(\cdot)$ is a weight function of the global mean intensity μ_k of $\mathbf{X}_k$ and the local mean intensity l_k of $\mathbf{x}_k$. $w_l(\cdot)$ is specified by a two dimensional Gaussian profile

$$w_l\left(\mu_k, l_k\right) = \exp\left(-\frac{(\mu_k-\tau)^2}{2\sigma_g^2} - \frac{(l_k-\tau)^2}{2\sigma_l^2}\right), \tag{11}$$

where σ_g and σ_l are two photometric spreads, set to 0.2 and 0.5, respectively [5]. $\tau = 128$ represents the mid-intensity value for an 8-bit sequence. The desired contrast is determined by the highest contrast in $\{\mathbf{x}_k\}$

$$\hat{c} = \max_{1\leq k\leq K} c_k\,. \tag{12}$$

The desired structure is computed by a weighted summation

$$\hat{\mathbf{s}} = \frac{\bar{\mathbf{s}}}{\|\bar{\mathbf{s}}\|}, \quad \text{where} \quad \bar{\mathbf{s}} = \frac{\sum_{k=1}^{K} w_s\left(\tilde{\mathbf{x}}_k\right)\mathbf{s}_k}{\sum_{k=1}^{K} w_s\left(\tilde{\mathbf{x}}_k\right)}, \tag{13}$$

where $w_s(\cdot) = \|\cdot\|_\infty$ is an ℓ_∞-norm weight function.

Once $\hat{l}$, $\hat{c}$, and $\hat{\mathbf{s}}$ are determined, we invert the decomposition to obtain the desired fused patch

$$\hat{\mathbf{x}} = \hat{c}\cdot\hat{\mathbf{s}} + \hat{l}\,. \tag{14}$$

The construction of MEF-SSIM follows the definition of the SSIM index [43]

$$S(\{\mathbf{x}_k\}, \mathbf{y}) = \frac{(2\mu_{\hat{\mathbf{x}}}\mu_{\mathbf{y}} + C_1)(2\sigma_{\hat{\mathbf{x}}\mathbf{y}} + C_2)}{(\mu_{\hat{\mathbf{x}}}^2 + \mu_{\mathbf{y}}^2 + C_1)(\sigma_{\hat{\mathbf{x}}}^2 + \sigma_{\mathbf{y}}^2 + C_2)}, \tag{15}$$

where $\mu_{\hat{\mathbf{x}}}$ and $\mu_{\mathbf{y}}$ denote the mean intensities of the desired patch and a given fused patch, respectively. $\sigma_{\hat{\mathbf{x}}}$, $\sigma_{\mathbf{y}}$, and $\sigma_{\hat{\mathbf{x}}\mathbf{y}}$ denote the local variances of $\hat{\mathbf{x}}$ and $\mathbf{y}$, and their covariance, respectively. C_1 and C_2 are two small positive constants to prevent instability. The local S values are averaged to obtain an overall quality measure of the fused image

$$\text{MEF-SSIM}(\{\mathbf{X}_k\}, \mathbf{Y}) = \frac{1}{M}\sum_{i=1}^{M} S(\{\mathbf{R}_i\mathbf{X}_k\}, \mathbf{R}_i\mathbf{Y}), \tag{16}$$

where $\mathbf{R}_i$ is a matrix that extracts the i-th patch from the image. The MEF-SSIM score ranges from 0 to 1 with a higher value indicating better visual quality.

The vanilla version of MEF-SSIM [9] excludes the intensity comparison and has been adopted by Prabhakar *et al.* [6] to drive the learning of convolutional networks for MEF. In our experiments, we find that optimizing MEF-SSIM without intensity information is unstable, resulting in fused images with a relatively pale appearance (see Fig. 6). The improved version of MEF-SSIM [8] adds the intensity comparison in Eq. (15) and directly works with color sequences. However, it is likely to generate over-saturated colors in some situations [8]. To obtain more conservative fused images with little artifacts, we choose to handle chroma components separately as suggested in [6]. Specifically, we work with the Y'CbCr format and evaluate MEF-SSIM only on the luma components of $\{\mathbf{X}_k\}$ and $\mathbf{Y}$. In other words, CAN in MEF-Net is optimized to fuse the luma components. For the Cb chroma components, we adopt a simple weighted summation suggested in [6]

$$\hat{b} = \frac{\sum_{k=1}^{K} w_c(b_k)b_k}{\sum_{k=1}^{K} w_c(b_k)}, \tag{17}$$

where b_k denotes the Cb chroma value at the k-th exposure and $w_c(b_k) = \|b_k - \tau\|_1$ is an ℓ_1-norm weight function. The Cr chroma components can be fused in the same way. Finally, we convert the fused image from Y'CbCr back to RGB.

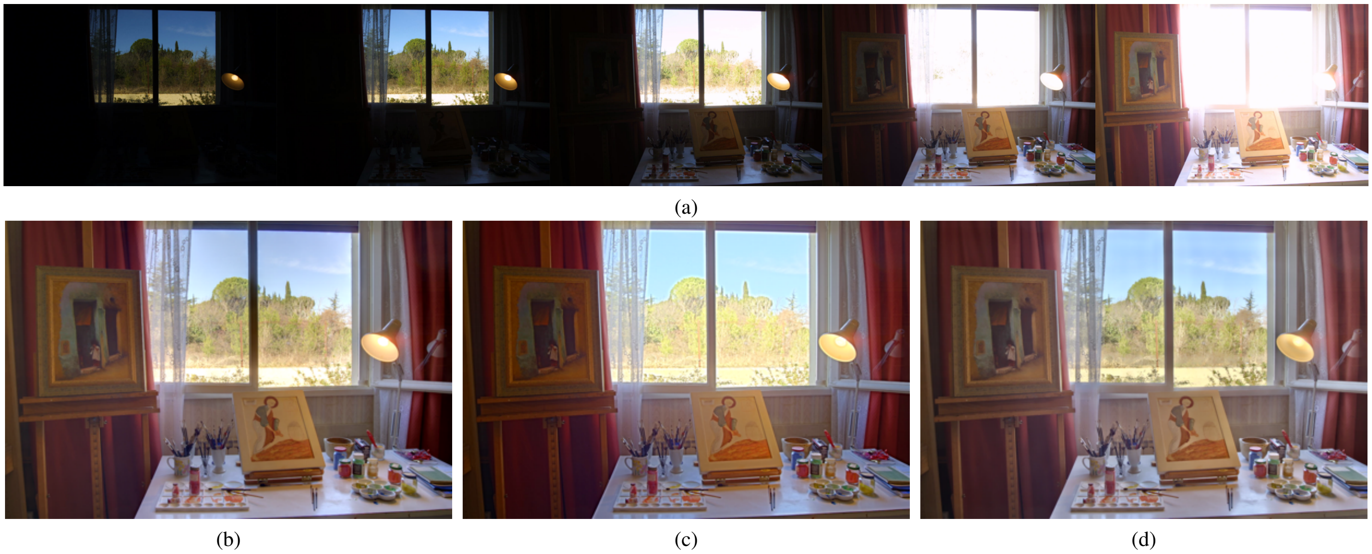


Fig. 4. MEF-Net in comparison with Mertens09 [2] and SPD-MEF [5]. (a) Source sequence "Studio" courtesy of HDRSoft. (b) Mertens09. (c) SPD-MEF. (d) MEF-Net.

D. Training

We collect a large-scale dataset for MEF-Net. Initially, we gather more than $1,000$ exposure sequences mainly from the five sources [6], [20]–[23]. We first eliminate sequences that contain visible object motion. For camera motion, we retain those sequences that have been successfully aligned by existing image registration algorithms [44]. After screening, a total of 690 static sequences remain, which span a great amount of HDR content, including indoor and outdoor, human and still-life, day and night scenes. Some representative sequences are shown in Fig. 3. The spatial resolution ranges from 0.2 to 20 megapixels, while the number of exposures is between three and nine. We train MEF-Net on 600 sequences and leave the remaining 90 for testing.

During training, we apply MEF-SSIM on the finest-scale only in order to reduce GPU memory cost. The parameters of MEF-SSIM are inherited from [8], [9]. We resize the exposure sequences to 128s and 512s as the low- and high-resolution inputs to MEF-Net, respectively, where 128s means that the short size is resized to 128 while keeping the aspect ratio. The leaky parameter λ_r of LReLU is fixed to 0.2. The radius r and the regularization parameter λ_a of the guided filter are set to 1 and 10^{-4}, respectively. λ_a is a critical parameter in MEF-Net, as will be clear in Section IV-B. Training uses the Adam solver [45] with a learning rate of 10^{-4}. Other parameters in Adam are set by default. The batch size is equal to the number of exposures in the current sequence. The learning stops when the maximum epoch number 100 is reached. We try to further train MEF-Net on sequences of varying high resolutions larger than 512s [19], but this does not yield noticeable improvement. Finally, we evaluate MEF-Net at full resolution during testing.

IV. Experiments

In this section, we first compare MEF-Net with classical and recent MEF methods in terms of visual quality and computational complexity. We then conduct a series of ablation experiments to identify the core components of MEF-Net. Last, we treat MEF-Net as a universal MEF approximator and use it to accelerate existing MEF methods.

A. Main Results

1) Qualitative Comparison: We compare MEF-Net with six previous MEF methods, including Mertens09 [2], Li13 [4], SPD-MEF [5], GGIF [7], DeepFuse [6], and MEF-Opt [8]. Mertens09 [2] is the primary baseline in MEF. Li13 [4] introduces guided filtering [18] to MEF, while GGIF [7] applies guided filtering in the gradient domain and achieves the best performance in a recent subjective experiment [20]. SPD-MEF is an MEF-SSIM-inspired non-iterative method and ranks second in the same subjective study [20]. MEF-Opt [8] is a gradient-based iterative method, optimizing MEF-SSIM [9] in the space of all images. DeepFuse [6] is a closely related method that trains a convolutional network for MEF. In principle, MEF-Opt can be regarded as an upper bound of all MEF methods in terms of MEF-SSIM. The fused images are generated by the implementations from the original authors with default settings. Since DeepFuse takes two exposures only, we try several under- and over-exposed combinations, and choose the fused image that achieves the best MEF-SSIM for comparison.

Fig. 4 compares Mertens09 [2] and SPD-MEF [5] with MEF-Net on the source sequence "Studio". As can be seen, Mertens09 does not recover the details of the lamp due to the extreme dynamic range of the scene and excessive Gaussian smoothing of the weight maps. In addition, the outside ground appears over-exposed. SPD-MEF does a good job in detail and color preservation of the indoor scene, but introduces annoying color and halo artifacts out of the window. We believe the distortions arise because SPD-MEF prefers strong or even over-saturated colors, whose weight maps fail to make smooth transitions across exposures near strong edges. By contrast,

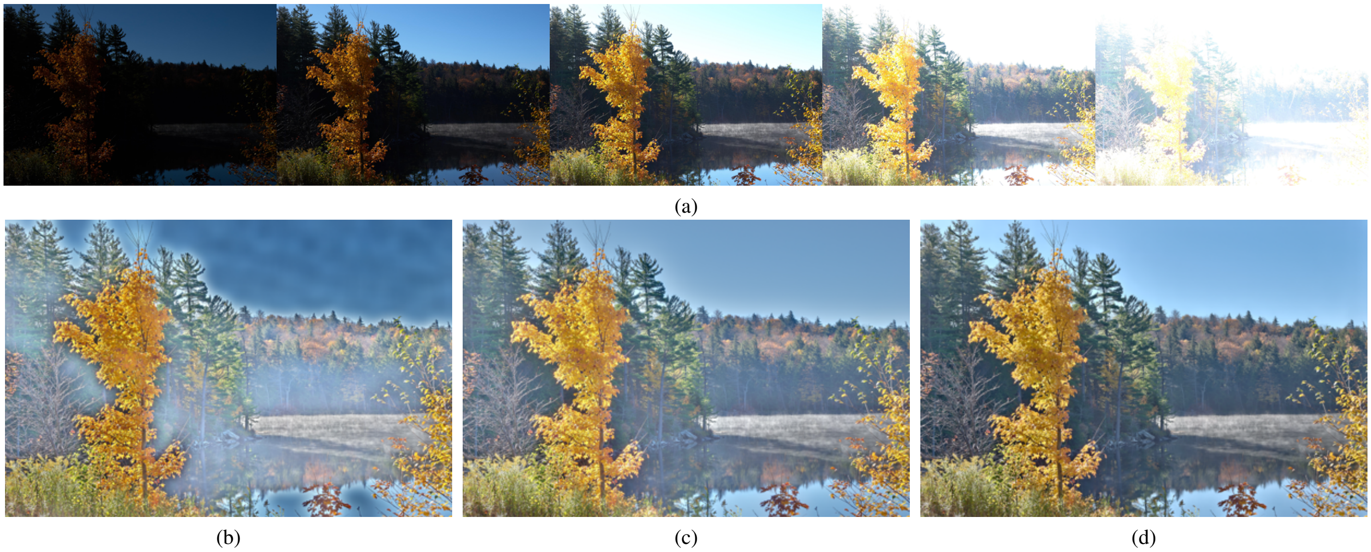


Fig. 5. MEF-Net in comparison with Li13 [4] and GGIF [7]. (a) Source sequence "Lake forest" courtesy of Jianrui Cai. (b) Li13. (c) GGIF. (d) MEF-Net.

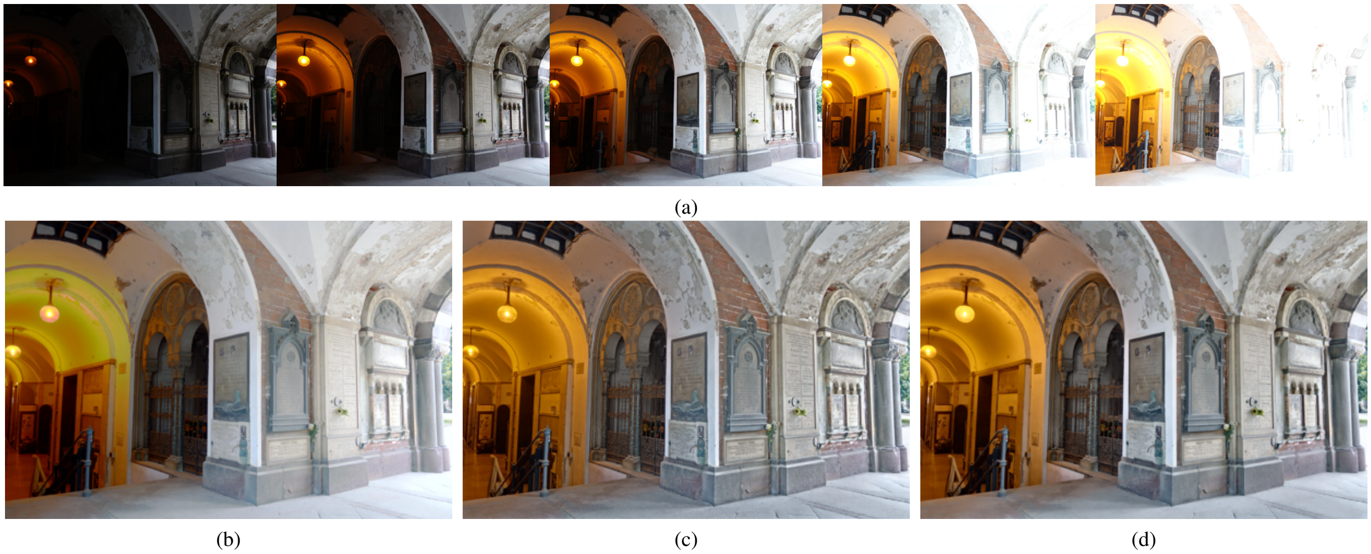
MEF-Net produces a more natural appearance with faithful detail and color reproduction.

Fig. 5 compares Li13 [4] and GGIF [7] with MEF-Net on the source sequence "Lake forest". By decomposing the input sequence into the base and detail layers with Gaussian filtering, Li13 focuses on fine-detail enhancement only and fails to capture large-scale luminance variations. Consequently, apparent halo artifacts emerge. Moreover, the global intensity of the fused image changes abruptly, resulting in an artificial and uncomfortable appearance. Inheriting the multi-scale Laplacian decomposition from Mertens09 [2], GGIF alleviates the halo artifacts to a just noticeable level, but at the same time reduces the global contrast. The fused image looks relatively pale and less detailed. Compared to GGIF, MEF-Net better preserves the global contrast, and the overall appearance of the fused image is more natural and appealing.

Fig. 6 compares DeepFuse [6] and MEF-Opt [8] with MEF-Net on the source sequence "Archway". The fusion performance of DeepFuse depends highly on the quality of the input image pair. If the under- and over-exposed images are not perfectly complementary, DeepFuse may generate a fused image of lower perceptual quality than a normally exposed shot. With only two exposures, it is difficult for DeepFuse to determine the lighting condition of the scene. The missing intensity component of MEF-SSIM during optimization makes the situation worse. As a result, we observe unnatural colors around the two lamps and reduced details on the wall and floor. By operating in the space of all images, MEF-Opt has more freedom than MEF-Net to produce the fused image with finer details, which is supported by a higher MEF-SSIM value. With a sensible network architecture, MEF-Net closely matches the result of MEF-Opt.

2) Quantitative Comparison: We list the quantitative comparison results in terms of MEF-SSIM [9] in Table II. It is not surprising that MEF-Opt [8] achieves the best performance because it optimizes MEF-SSIM in the space of all images. Among the rest of the methods, MEF-Net is closest to this upper bound, which suggests that the training is highly effective, and MEF-Net generalizes well to novel content. Although sharing the same spirit of MEF-SSIM optimization, DeepFuse [6] performs the worst due to the extremely strict constraint on the input sequence. We also employ another subject-calibrated quality model specifically for MEF, namely MEF-VIF [42], to quantify the fusion performance on the same 90 test sequences. From Table II, we see that MEF-Net is among the best performing methods. The proposed MEF-Net is flexible and may be trained to optimize MEF-VIF directly.

We take a closer look at the cross resolution generalizability of MEF-Net. Specifically, we downsample the 90 test sequences to seven resolutions if possible, ranging from 512s to 2048s, and report the average MEF-SSIM scores in Fig. 7. Despite the fact that MEF-Net is trained on the resolution of 512s, it generalizes remarkably well across a wide range of unseen resolutions with slight MEF-SSIM decrease. Meanwhile, we observe a steady uptrend of MEF-Opt [8] optimized for MEF-SSIM with the increasing resolution. This may arise because for most MEF algorithms including MEF-Opt, it is easier to fuse flat regions than structured ones; when the spatial resolution increases, the flat regions grow more rapidly than the structured regions (consider the step-edge images of different sizes). Other MEF methods perform equally well except for Mertens09 [2], which is not scale-invariant. Mertens09 employs Laplacian pyramid [24] to avoid unwanted artifacts during fusion. The standard implementation of Laplacian pyramid uses a 5×5 lowpass filter, which may not eliminate high-frequency information before downsampling (by a factor of two), leading to possible aliasing artifacts across scales. Therefore, we may only observe scale-invariance when the image resolutions are related by multipliers of two, which is verified by approximately the same MEF-SSIM scores computed at 512s, 1024s, and 2048s in Fig. 7. By replacing Gaussian filtering with guided filtering, GGIF [7] achieves the desired scale-invariance within the same framework.



(a)

(b) (c) (d)

Fig. 6. MEF-Net in comparison with DeepFuse [6] and MEF-Opt [8]. (a) Source sequence "Archway" courtesy of Jianrui Cai. (b) DeepFuse. (c) MEF-Opt. (d) MEF-Net.

TABLE II
AVERAGE MEF-SSIM [9] AND MEF-VIF [42] SCORES OF DIFFERENT MEF METHODS AGAINST MEF-NET ON 90 TEST SEQUENCES COMPUTED AT FULL RESOLUTION. BOTH MEF-SSIM AND MEF-VIF SCORES RANGE FROM 0 TO 1 WITH A HIGHER VALUE INDICATING BETTER PERCEPTUAL QUALITY

MEF method	Mertens09 [2]	Li13 [4]	SPD-MEF [5]	GGIF [7]	DeepFuse [6]	MEF-Opt [8]	MEF-Net
MEF-SSIM [9]	0.923	0.945	0.953	0.958	0.862	**0.978**	**0.964**
MEF-VIF [42]	**0.969**	0.967	0.956	**0.972**	0.926	0.952	0.967

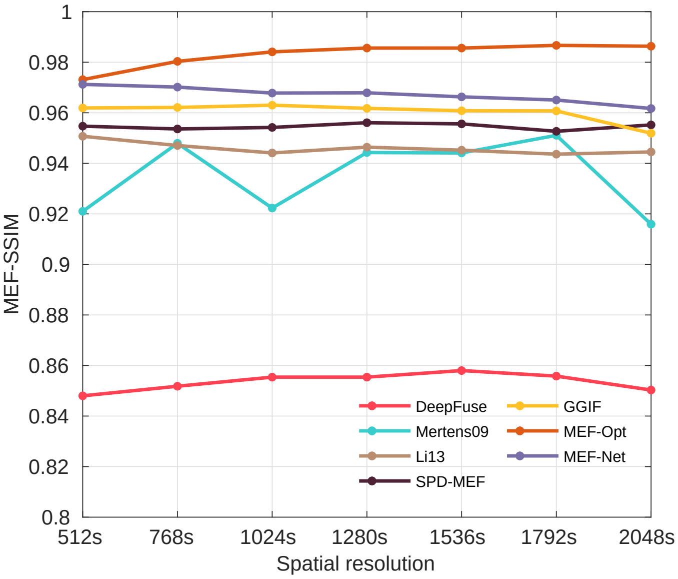
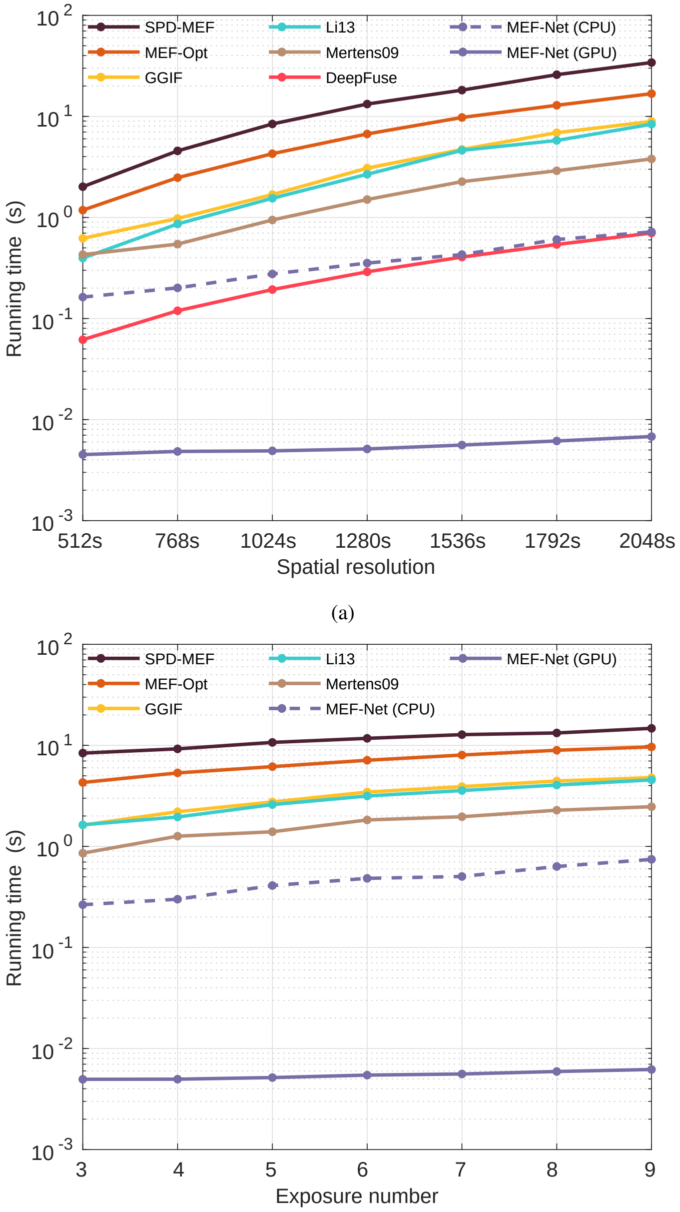


Fig. 7. Cross resolution generalization. MEF-Net generalizes well across a wide range of resolutions, which are never seen during training.

3) Computational Complexity and Running Time: We conduct a computational complexity comparison of MEF methods in terms of the number of floating point operations. We assume that the number of input channels is K, each of which contains M pixels, and the window size used to compute local statistics is N^2. All competing MEF algorithms have a complexity of $\mathcal{O}(KMN^2)$, except for MEF-Opt [8] which has a complexity of $\mathcal{O}(IKMN^2)$, where I is the number of iterations. Ideally, the computation across the channel dimension can be parallelized and the value of K should have little impact on the running time (given sufficient code optimization). Due to the fact that $N^2 \ll M$, the spatial resolution M is the dominant term. MEF-Net enjoys the lowest computational complexity because it restricts most of the computation at a fixed low resolution, while the competing MEF algorithms need to perform all computation at full resolution.

We compare the running time of MEF-Net with existing MEF methods on input sequences of different spatial resolutions or different numbers of exposures. The testing platform is a computer with an Intel i7-6900K 3.2GHz CPU and an Nvidia Titan X GPU. Mertens09 [2], Li13 [4], SPD-MEF [5], and GGIF [7] utilize CPU, while DeepFuse [6] and MEF-Opt [8] exploit GPU. We do not report the running time of DeepFuse on sequences of different numbers of exposures due to its strict input constraint. We reduce the maximum iteration number of MEF-Opt to 100 for the ease of drawing. The results are shown in Fig. 8. On the GPU, MEF-Net takes less than 10 ms to process sequences with resolutions ranging from 512s to 2048s and exposure numbers ranging from three to nine, which is 10× and 1000× faster than DeepFuse and SPD-MEF, respectively. More importantly, MEF-Net runs in approximately constant time in spite of the growing spatial resolution and the number of exposures. On CPU, MEF-Net is still significantly faster than most MEF methods except for the GPU-mode DeepFuse.



(a)

(b)

Fig. 8. Running time comparison. (a) Different spatial resolutions with the number of exposures fixed to 3. (b) Different numbers of exposures with the spatial resolution fixed to 1024s.

TABLE III
AVERAGE MEF-SSIM [9] SCORES OF MEF-NET AND ITS VARIANTS

NEF-Net variants	MEF-SSIM [9]
Guided filtering as post-processing	0.953
Bilinear upsampling trained end-to-end	0.961
MEF-Net (guided filtering trained end-to-end)	**0.964**

In summary, we have empirically shown that the proposed MEF-Net, characterized by CAN and guided filtering, trained end-to-end, achieves the three desirable properties—*flexibility*, *speed*, and *quality*—in MEF.

TABLE IV
AVERAGE MEF-SSIM [9] SCORES AS A FUNCTION OF INPUT RESOLUTION, DEPTH, AND WIDTH OF CAN IN MEF-NET. THE DEFAULT SETTING IS HIGHLIGHTED IN BOLD

Input res		32	64	**128**	256	
MEF-SSIM		0.950	0.960	0.964	**0.967**	
Depth	4	5	6	**7**	8	9
MEF-SSIM	0.961	0.963	0.963	0.964	**0.965**	**0.965**
Width	8	16	**24**	32	48	64
MEF-SSIM	0.953	0.963	0.964	0.966	**0.967**	**0.967**

TABLE V
AVERAGE MEF-SSIM [9] SCORES AS A FUNCTION OF THE REGULARIZATION PARAMETER λ_a AND THE RADIUS r IN THE GUIDED FILTER. THE DEFAULT SETTING IS HIGHLIGHTED IN BOLD

λ_a	10^{-1}	10^{-2}	$\mathbf{10^{-4}}$	10^{-6}	10^{-8}
MEF-SSIM	0.961	0.961	**0.964**	0.962	0.961
r	**1**	2	4	8	16
MEF-SSIM	**0.964**	0.963	0.959	0.956	0.950

B. Ablation Experiments

We conduct comprehensive ablation experiments to single out the contribution of each component in MEF-Net. We first train MEF-Net on low-resolution sequences solely. After training, the guided filter is adopted as a post-processing step to jointly upsample the low-resolution weight maps for final fusion. We then train MEF-Net with the guided filter replaced by the simple bilinear upsampler. The MEF-SSIM [9] results are listed in Table III, where we see that integrating upsampling techniques with the preceding CAN for end-to-end training significantly boosts MEF-SSIM. This verifies the power of end-to-end training, where MEF-Net is directly supervised by the high-resolution input sequences. Additional performance gain can be obtained by guided filtering over bilinear upsampling. We also provide a visual demonstration in Fig. 9 and find that guided filtering as post-processing exhibits over-exposure out of the window, while bilinear upsampling trained end-to-end shows black banding artifacts due to the excessively coarse weight maps. Guided filtering trained end-to-end for joint upsampling achieves the best visual quality, and is the key component of MEF-Net.

We next evaluate the effect of input resolution, depth, and width of CAN on the performance of MEF-Net. The depth and width represent the number of convolution layers and the number of feature maps in each intermediate layer, respectively. A shallower CAN implies a smaller receptive field. The results are listed in Table IV, from which we have several interesting observations. First, MEF-SSIM increases with input resolution, depth, and width as expected. Second, by changing the input resolution from 128s to 256s, we observe marginal MEF-SSIM improvement by 0.003. Third, MEF-Net achieves satisfactory performance with a fairly shallow and compact architecture (*e.g.*, with 16 feature maps per layer or a depth of five).

We also assess the role of the regularization parameter λ_a and the radius r in the guided filter. λ_a controls the smoothness of $\mathbf{A}_k$, which is evident in Eq. (6). r also affects

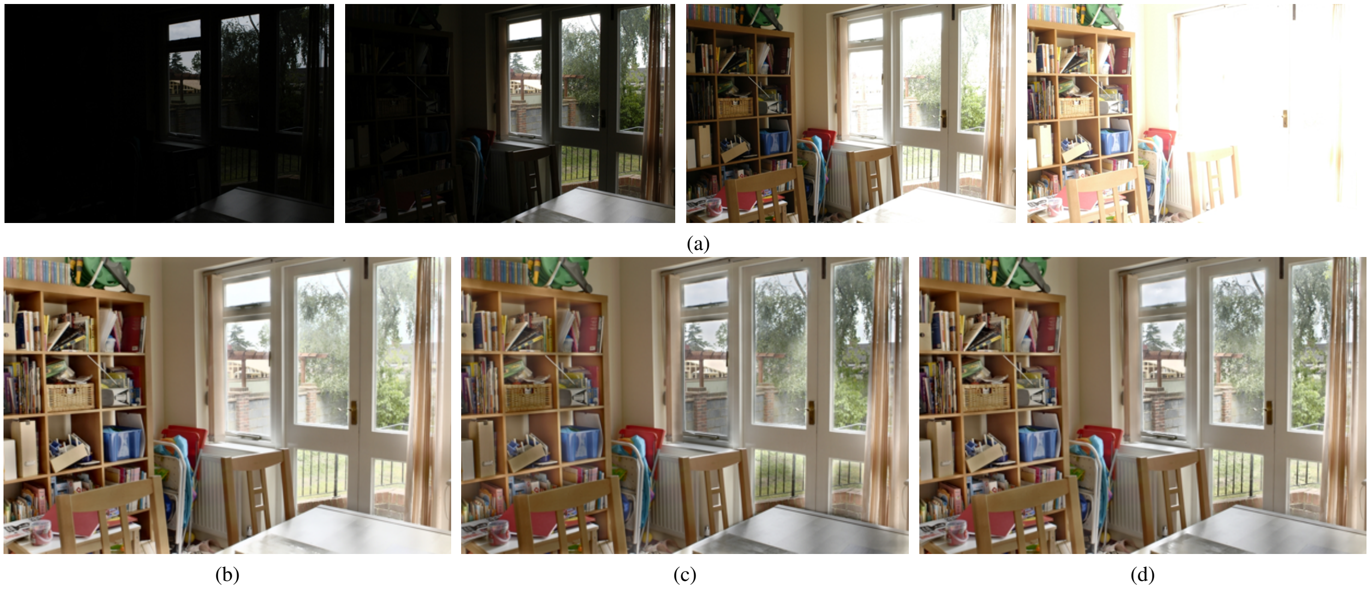


Fig. 9. MEF-Net in comparison with its variants. (a) Source sequence “House” courtesy of Tom Mertens. (b) Guided filtering as post-processing. (c) Bilinear upsampling trained end-to-end. (d) MEF-Net (guided filtering trained end-to-end).

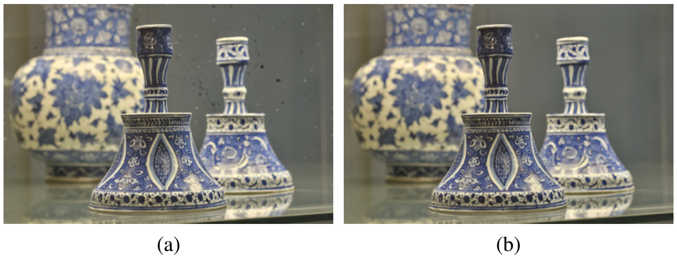


Fig. 10. Emergence of the dot artifacts with a small regularization parameter λ_a in the guided filter. (a) $\lambda_a = 10^{-8}$. (b) $\lambda_a = 10^{-4}$ (default).

the smoothness of $\mathbf{A}_k$ in a less direct way. A large λ_a (or r) generates a smooth $\hat{\mathbf{W}}_k$ and may not be good at preserving fine details, leading to a decrease of MEF-SSIM in Table V. A small λ_a produces a relatively noisy $\hat{\mathbf{W}}_k$ and may introduce dot artifacts, as shown in Fig. 10. Our default setting achieves the best performance.

C. MEF-Net as A Universal MEF Approximator

In this subsection, we exploit the fast speed of MEF-Net and use it as a universal approximator to accelerate existing MEF methods. Specifically, we first apply the target MEF method to our dataset. The generated fused images are considered as the ground truths. We then train MEF-Net on input/output pairs that contain the exposure sequences and the corresponding fused images. The training procedure is the same as Section III-D, except that we optimize a perceptual image quality metric—SSIM [43] in the RGB space. We have also experimented with the mean squared error (MSE) suggested in [14], [19], but obtain inferior approximation accuracy.

Fig. 11 shows the visual results of MEF-Net in approximating SPD-MEF [5] and GGIF [7] on the source sequence “Stone house”. Although the two MEF methods produce different overall appearances, MEF-Net is able to closely match them. On the 90 test sequences, the approximation accuracy in terms of SSIM for SPD-MEF and GGIF is 0.961 and 0.976, respectively, demonstrating the promise of MEF-Net as a universal MEF approximator. On sequences of resolution 1024s, we speed up SPD-MEF and GGIF more than 1000 and 100 times, respectively.

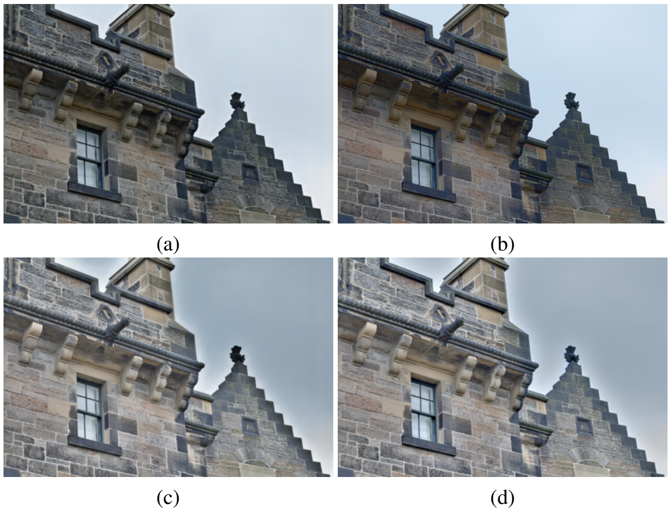


Fig. 11. MEF-Net as a universal approximator. (a) SPD-MEF [5]. (b) SPD-MEF approximated by MEF-Net. (c) GGIF [7]. (d) GGIF approximated by MEF-Net. Source sequence “Stone house” courtesy of Jianrui Cai.

V. Conclusion and Discussion

We have introduced MEF-Net, a fast MEF method based on deep guided learning. The core idea of MEF-Net is to predict the low-resolution weight maps using a CAN and jointly upsample them with a guided filter for final weighted fusion.

The high speed of MEF-Net is achieved by restricting the main computation at a fixed low resolution and parallelizing the computation across exposures. The visual improvement of the fused images is achieved by end-to-end training with MEF-SSIM measured at full resolution. In addition, we demonstrate the promise of MEF-Net as a universal MEF approximator to accelerate existing and future MEF methods.

The current MEF-Net works with static scenes only. How to extend it to account for dynamic scenes is an interesting and challenging problem yet to be explored. The major impediment here is the lack of perceptual image quality metrics [46] for dynamic scenes or ground truths for supervision. Kalantari and Ramamoorthi [16] put substantial effort in capturing static and dynamic exposure brackets of the same scene and treated the static sequences as a form of ground truths. Cai *et al*. [20] made use of 13 existing MEF and HDR deghosting methods to generate a set of candidates and manually picked the best ones as the ground truths. Both processes are expensive and time-consuming, which limit the number of collected sequences. In addition, we desire more flexible and faster MEF methods for dynamic scenes.

Acknowledgements

The authors would like to thank Jianrui Cai, Yuki Endo, K Ram Prabhakar, Dr. Nima Kalantari and Dr. Mark Fairchild for providing their multi-exposure sequences, and Hui Li for fruitful discussions. We thank NVIDIA Corporation for donating a GPU for this research.

References

[1] E. Reinhard, W. Heidrich, P. Debevec, S. Pattanaik, G. Ward, and K. Myszkowski, *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*. Morgan Kaufmann, 2010.

[2] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion: A simple and practical alternative to high dynamic range photography," *Computer Graphics Forum*, vol. 28, no. 1, pp. 161–171, Feb. 2009.

[3] Z. Li, J. Zheng, and S. Rahardja, "Detail-enhanced exposure fusion," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4672–4676, Nov. 2012.

[4] S. Li, X. Kang, and J. Hu, "Image fusion with guided filtering," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2864–2875, Jul. 2013.

[5] K. Ma, H. Li, H. Yong, Z. Wang, D. Meng, and L. Zhang, "Robust multi-exposure image fusion: A structural patch decomposition approach," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2519–2532, May 2017.

[6] K. R. Prabhakar, V. S. Srikar, and R. V. Babu, "Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs," in *IEEE International Conference on Computer Vision*, 2017, pp. 4724–4732.

[7] F. Kou, Z. Li, C. Wen, and W. Chen, "Multi-scale exposure fusion via gradient domain guided image filtering," in *IEEE International Conference on Multimedia and Expo*, 2017, pp. 1105–1110.

[8] K. Ma, Z. Duanmu, H. Yeganeh, and Z. Wang, "Multi-exposure image fusion by optimizing a structural similarity index," *IEEE Transactions on Computational Imaging*, vol. 4, no. 1, pp. 60–72, Mar. 2018.

[9] K. Ma, K. Zeng, and Z. Wang, "Perceptual quality assessment for multi-exposure image fusion," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3345–3356, Nov. 2015.

[10] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time $O(1)$ bilateral filtering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 557–564.

[11] J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, and F. Durand, "Decoupling algorithms from schedules for easy optimization of image processing pipelines," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 32:1–32:12, Jul. 2012.

[12] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 192:1–192:12, Nov. 2016.

[13] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 118:1–118:12, Jul 2017.

[14] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2516–2525.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[16] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 144:1–144:12, Jul. 2017.

[17] S. Wu, J. Xu, Y. Tai, and C. Tang, "End-to-end deep HDR imaging with large foreground motions," *CoRR*, vol. abs/1711.08937, 2017. [Online]. Available: http://arxiv.org/abs/1711.08937

[18] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[19] H. Wu, S. Zheng, J. Zhang, and K. Huang, "Fast end-to-end trainable guided filter," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1838–1847.

[20] J. Cai, S. Gu, and L. Zhang, "Learning a deep single image contrast enhancer from multi-exposure images," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2049–2062, Jan. 2018.

[21] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 177:1–177:10, Nov. 2017.

[22] M. Fairchild, "The HDR photographic survey," 2008 (accessed Apr. 18, 2018). [Online]. Available: http://rit-mcsl.org/fairchild/HDR.html

[23] K. Zeng, K. Ma, R. Hassen, and Z. Wang, "Perceptual evaluation of multi-exposure image fusion algorithms," in *Sixth IEEE International Workshop on Quality of Multimedia Experience*, 2014, pp. 7–12.

[24] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, Apr. 1983.

[25] P. J. Burt and R. J. Kolczynski, "Enhanced image capture through fusion," in *IEEE International Conference on Computer Vision*, 1993, pp. 173–182.

[26] S. Li, X. Kang, L. Fang, J. Hu, and H. Yin, "Pixel-level image fusion: A survey of the state of the art," *Information Fusion*, vol. 33, pp. 100–112, Jan. 2017.

[27] A. A. Goshtasby, "Fusion of multi-exposure images," *Image and Vision Computing*, vol. 23, no. 6, pp. 611–618, Jun. 2005.

[28] K. Ma and Z. Wang, "Multi-exposure image fusion: A patch-wise approach," in *IEEE International Conference on Image Processing*, 2015, pp. 1717–1721.

[29] W. Zhang and W.-K. Cham, "Gradient-directed multiexposure composition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2318–2323, Apr. 2012.

[30] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 257–266, Jul. 2002.

[31] B. Weiss, "Fast median and bilateral filtering," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 519–526, Jul. 2006.

[32] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 103:1–103:10, Jul. 2007.

[33] M. Gharbi, Y. Shih, G. Chaurasia, J. Ragan-Kelley, S. Paris, and F. Durand, "Transform recipes for efficient cloud photo enhancement," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 228:1–228:12, Nov. 2015.

[34] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations*, 2016.

[35] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, "Fast local Laplacian filters: Theory and applications," *ACM Transactions on Graphics*, vol. 33, no. 5, pp. 167:1–167:14, Aug. 2014.

[36] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 67:1–67:10, Aug. 2008.

[37] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 96:1–96:6, Jul. 2007.

[38] K. Ma, H. Fu, T. Liu, Z. Wang, and D. Tao, "Deep blur mapping: Exploiting high-level semantics by deep neural networks," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 5155–5166, Jun. 2018.

[39] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *CoRR*, vol. abs/1607.08022, 2016. [Online]. Available: http://arxiv.org/abs/1607.08022

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[41] Z. Liu, E. Blasch, Z. Xue, J. Zhao, R. Laganiere, and W. Wu, "Objective assessment of multiresolution image fusion algorithms for context enhancement in night vision: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 94–109, Jan. 2012.

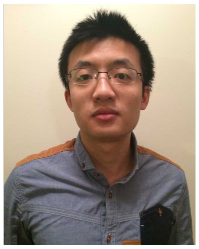
[42] H. Rahman, R. Soundararajan, and R. V. Babu, "Evaluating multiexposure fusion using image information," *IEEE Signal Processing Letters*, vol. 24, no. 11, pp. 1671–1675, Nov. 2017.

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

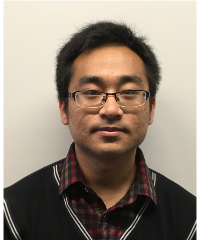
[44] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[46] Y. Fang, H. Zhu, K. Ma, Z. Wang, and S. Li, "Perceptual evaluation for multi-exposure image fusion of dynamic scenes," *IEEE Transactions on Image Processing*, to appear, 2019.



Kede Ma (S'13-M'18) received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2012, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2014 and 2017, respectively. He was a Research Associate with Howard Hughes Medical Institute and New York University, New York, NY, USA, in 2018. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. His research interests include perceptual image processing, computational vision, and computational photography.



Zhengfang Duanmu (S'15) received the B.A.Sc. and M.A.Sc. degrees from the University of Waterloo in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include adaptive streaming, perceptual image processing, and quality of experience.

Hanwei Zhu (S'17) received the B.E. degree from the Jiangxi University of Finance and Economics, Nanchang, China, in 2017, where he is currently pursuing the M.S. degree in computer science. His research interest lies in perceptual image processing.

Yuming Fang (M'13-SM'17) received the B.E. degree from Sichuan University, Chengdu, China, the M.S. degree from the Beijing University of Technology, Beijing, China, and the Ph.D. degree from Nanyang Technological University, Singapore. He is currently a Professor with the School of Information Management, Jiangxi University of Finance and Economics, Nanchang, China. His research interests include visual attention modeling, visual quality assessment, computer vision, and 3D image/video processing. He serves as an Associate Editor for IEEE ACCESS. He is on the editorial board of *Signal Processing: Image Communication*.

Zhou Wang Zhou Wang (S'99-M'02-SM'12-F'14) received the Ph.D. degree from The University of Texas at Austin in 2001. He is currently a Canada Research Chair and Professor in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include image and video processing and coding; visual quality assessment and optimization; computational vision and pattern analysis; multimedia communications; and biomedical signal processing. He has more than 200 publications in these fields with over 50,000 citations (Google Scholar).

Dr. Wang serves as a member of IEEE IMAGE, VIDEO AND MULTIDIMENSIONAL SIGNAL PROCESSING TECHNICAL COMMITTEE (2020-2022) and IEEE MULTIMEDIA SIGNAL PROCESSING TECHNICAL COMMITTEE (2013-2015), a Senior Area Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING (2015-2019), and an Associate Editor or Guest Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2016-2018), IEEE TRANSACTIONS ON IMAGE PROCESSING (2009-2014), IEEE SIGNAL PROCESSING LETTERS (2006-2010), and IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (2013-2014 and 2007-2009), among other journals. He was elected a Fellow of Royal Society of Canada, Academy of Science in 2018, and a Fellow of Canadian Academy of Engineering in 2016. He is a recipient of 2016 IEEE Signal Processing Society Sustained Impact Paper Award, 2015 Primetime Engineering Emmy Award, 2014 NSERC E.W.R. Steacie Memorial Fellowship Award, 2013 IEEE Signal Processing Magazine Best Paper Award, and 2009 IEEE Signal Processing Society Best Paper Award.