This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

MEFLUT: Unsupervised 1D Lookup Tables for Multi-exposure Image Fusion

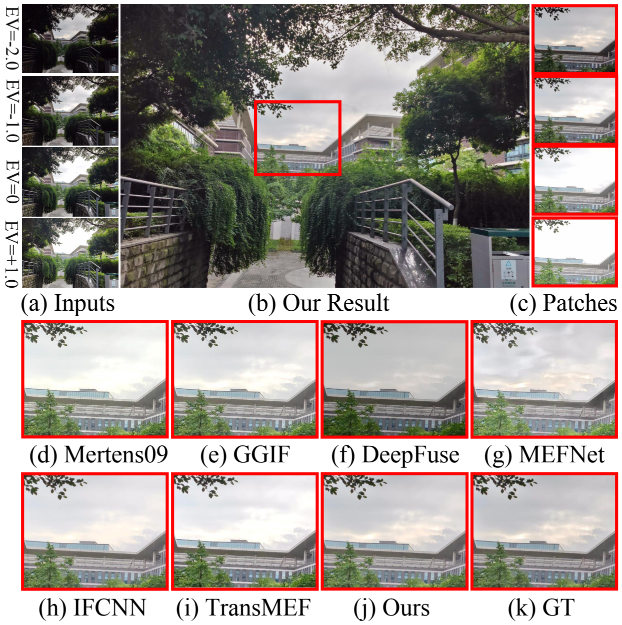
Ting Jiang [1] Chuan Wang [1] Xinpeng Li [1] Ru Li [1] Haoqiang Fan [1] Shuaicheng Liu [2,1]*
[1] Megvii Technology
[2] University of Electronic Science and Technology of China

Abstract

In this paper, we introduce a new approach for high-quality multi-exposure image fusion (MEF). We show that the fusion weights of an exposure can be encoded into a 1D lookup table (LUT), which takes pixel intensity value as input and produces fusion weight as output. We learn one 1D LUT for each exposure, then all the pixels from different exposures can query 1D LUT of that exposure independently for high-quality and efficient fusion. Specifically, to learn these 1D LUTs, we involve attention mechanism in various dimensions including frame, channel and spatial ones into the MEF task so as to bring us significant quality improvement over the state-of-the-art (SOTA). In addition, we collect a new MEF dataset consisting of 960 samples, 155 of which are manually tuned by professionals as ground-truth for evaluation. Our network is trained by this dataset in an unsupervised manner. Extensive experiments are conducted to demonstrate the effectiveness of all the newly proposed components, and results show that our approach outperforms the SOTA in our and another representative dataset SICE, both qualitatively and quantitatively. Moreover, our 1D LUT approach takes less than 4ms to run a 4K image on a PC GPU. Given its high quality, efficiency and robustness, our method has been shipped into millions of Android mobiles across multiple brands world-wide. Code is available at: https://github.com/Hedlen/MEFLUT.

1. Introduction

Dynamic ranges of natural scenes are much wider than those captured by commercial imaging products, causing they are hard to be captured by most digital photography sensors. As a result, high dynamic range (HDR) [1, 2] imaging techniques have attracted considerable interest due to their capability to overcome such limitations. Among HDR solutions, multi-exposure image fusion (MEF) [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] provides a cost-effective one, with which plausible images with vivid detail can be generated. MEF has attracted wide attention and there have been many methods [16, 17, 18, 19, 20, 21, 22, 23, 24] available to fuse images with faithful detail and color reproduction. However, these MEF methods use hand-crafted features or transformations so that they usually suffer from robustness if being applied to modified conditions.



(a) Inputs (b) Our Result (c) Patches
(d) Mertens09 (e) GGIF (f) DeepFuse (g) MEFNet
(h) IFCNN (i) TransMEF (j) Ours (k) GT

Figure 1. (a) Inputs. (b) Our result. (c) Input patches. We show the results by various methods in patches. Our result hallucinates more accurate details over the saturated areas.

Inspired by the successes of the deep neural networks (DNNs) in many computer vision areas [25, 26, 27, 28], recently some deep learning-based approaches [5, 7] have been proposed to improve MEF. DeepFuse [5] uses DNNs for the first time to directly regress the Y channel of an YUV image as the target. As the core weight maps' learning is ignored, its quality is limited despite of its acceptable efficiency due to the limited model size. To improve the quality, MEFNet [7] alternatively learns the weight maps for blending the input sequence. However, its speed is downgraded as the network becomes complex, hence it just works on low resolution input as a workaround to maintain the efficiency. Unfortunately, these methods do not take real-world deployment into consideration, their speed and quality cannot be well balanced, causing the difficulty of their wide

*Corresponding author

application such as into mobile platform. To tackle these issues, we propose a method named MEFLUT, which aims to achieve higher quality and efficiency simultaneously by taking advantages of deep learning techniques. Our method has no strict requirements on the running platform, it can run in PC and mobile CPU and GPU. As shown in Figs. 1 and 7, our method outperforms all the other methods in the image detail preservation and running speed.

MEFLUT consists of two parts. Firstly, we design a network based on multi-dimensional attention mechanism, which is trained via unsupervised methods. The attention mechanism works in frame, channel and spatial dimension separately to fuse inter-frame and intra-frame features, which brings us quality gain in detail preservation. After the network converges, we simplify the model to multiple 1D LUTs, that encodes the fusion weight for a given input pixel value from a specific exposure image. In the test phase, the fusion weights corresponding to different exposures are directly queried from the LUTs. In order to further accelerate our method, the input is downsampled to obtain the fusion masks, which are then upsampled to the original resolution for the fusion, by guided filtering for upsampling (GFU) [7] to avoid boundary stratification. We verify that with or without GFU, our method always runs faster than our competitors, revealing the key effect of LUTs learned.

Besides, considering none of the existing MEF datasets is completely collected through mobile devices, we therefore build a high-quality multi-exposure image sequence dataset. Specifically, we spent over one month to capture and filter out 960 multi-exposure images covering a diversity of scenes by different brands of mobile phones. Among them, 155 samples are produced with ground-truth (GT) for the purpose of quantitative evaluation, which are produced by first running image predictions by 14 algorithms, voted by 20 volunteers and then fine-tuned by an Image Quality (IQ) expert. Producing these GT samples totally cost at least 40 man-hours without counting the organization effort.

To sum up, our main contributions include:

- We propose MEFLUT that learns 1D LUTs for the task of MEF. We show that the fusion weights can be encoded into the LUTs successfully. Once learned, MEFLUT can be easily deployed with high efficiency, so that a 4K image runs in less than 4ms on a PC GPU. To the best of our knowledge, this is the first time to demonstrate the benefits of LUTs for MEF.
- We propose a new network structure with two attention modules in all dimensions that outperforms the state-of-the-art in quality especially detail preservation.
- We also release a new dataset of 960 multi-exposure image sequences collected by mobile phones in various brands and from diverse scenes. 155 samples of them are produced detailed image as ground-truth target by professionals manually.

2. Related Works

2.1. Existing MEF Algorithms

MEF tasks typically perform as a weighted summation of multiple frames with different exposures. Therefore, the focus of MEF is often to find an appropriate method to get the weights of different exposures. Mertens *et al.* [17] uses contrast, saturation and well-exposeness of each exposure to get the fused weights. Compared with these traditional MEF methods [29, 3, 17, 30, 20, 31, 22], which focus on obtaining the weight in advance, some others [5, 32, 24, 8, 9, 13, 14, 15] prefer to transform the MEF task into an optimization problem. Ma *et al.* [23] proposed a gradient based method to minimize MEF-SSIM for search better fusion results in image space. However, this method requires searching in each fusion causing that it is so time-consuming. In recent years, some deep learning methods also try to optimize the model through MEF-SSIM. For example, DeepFuse [5] accomplishes the MEF task via a neural network, achieving faster computation than the traditional method while keeping the fusion quality. Recently, Zhang *et al.* [8] proposed a general image fusion framework IFCNN, which is based on DNNs and direct reconstructs the fusion results through the network. Qu *et al.* [14] proposed the TransMEF, which uses a Transformer to further improve the quality of MEF. However, these methods do not consider speed and are not designed for mobile devices.

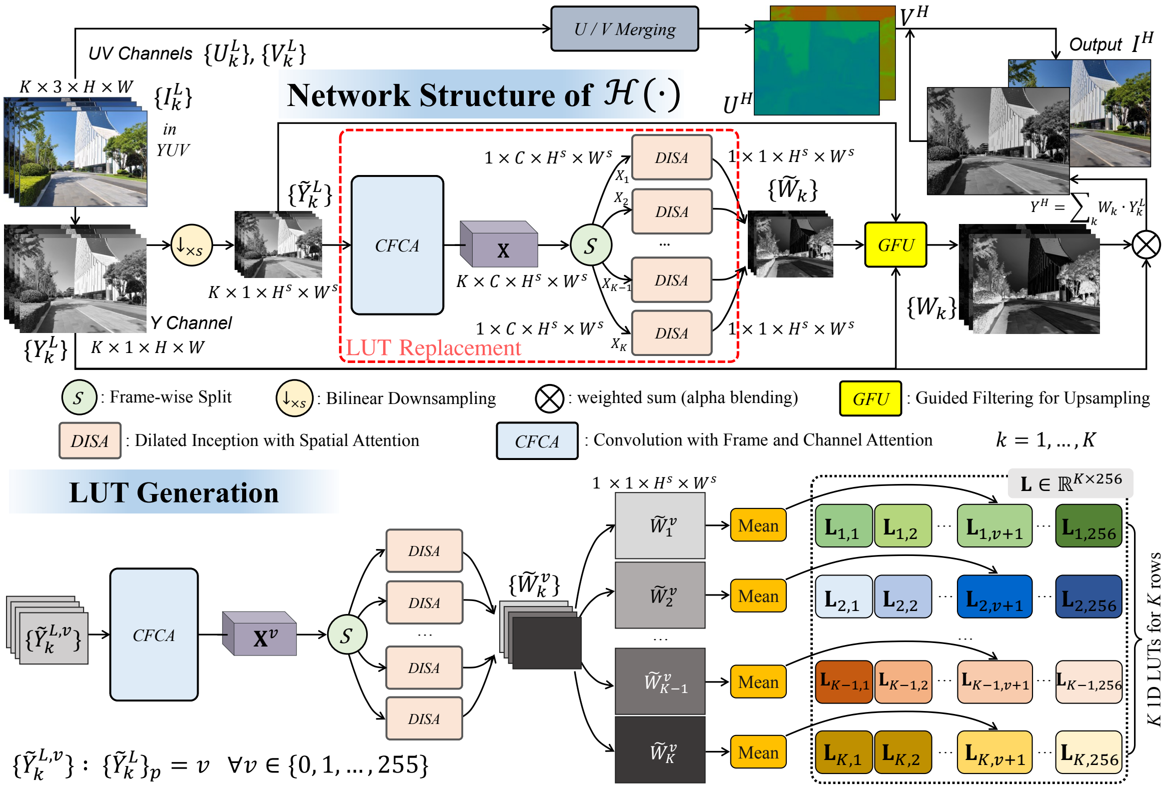
2.2. Acceleration of MEF Algorithms

Most of the aforementioned methods are time-consuming considering potential deployment in mobile platform, as mobile devices are relatively of limited computing power. One solution is to cloud-based solution while for high resolution images, image transmission is also time-consuming. Another solution is to conduct the computation in down-sampled version of images like [33, 34, 35, 36, 37, 7]. MEFNet [7] used guided filter [38] to realize upsampling, which can well preserve the high-frequency and edge information. However, this method is too complicated and time-consuming even on small resolution images. Therefore, we propose a new MEF method based on LUT for the efficient and high quality fusion.

Another problem faced by MEF task on mobile devices is that there lack of datasets captured by causal cameras, although there exist public datasets such as SICE [39] and HDREYE [40] taken by professional cameras, which means the images is high-quality. The trained model by these datasets may be hard to be applied to mobile devices due to the generalization problem. Therefore, we propose a comprehensive dataset to broaden the applications.

2.3. LUT

The LUT has been widely used in vision tasks, including image enhancement [41, 42], super-resolution [43, 44]



UV Channels $\{U_k^L\}, \{V_k^L\}$
U / V Merging
V^H
Output I^H
U^H
Network Structure of $\mathcal{H}(\cdot)$
$K \times 3 \times H \times W$
$\{I_k^L\}$
in
YUV
$\{\tilde{Y}_k^L\}$
$\downarrow_{\times s}$
$K \times 1 \times H^s \times W^s$
Y Channel
$\{Y_k^L\}$
$K \times 1 \times H \times W$
CFCA
X
$K \times C \times H^s \times W^s$
S
$1 \times C \times H^s \times W^s$
X_1
X_2
X_{K-1}
X_K
DISA
DISA
DISA
DISA
...
$1 \times 1 \times H^s \times W^s$
$\{\tilde{W}_k\}$
LUT Replacement
GFU
$\{W_k\}$
$Y^H = \sum_k W_k \cdot Y_k^L$
S : Frame-wise Split
$\downarrow_{\times s}$: Bilinear Downsampling
: weighted sum (alpha blending)
GFU : Guided Filtering for Upsampling
DISA : Dilated Inception with Spatial Attention
CFCA : Convolution with Frame and Channel Attention
$k = 1, \ldots, K$
LUT Generation
$\{\tilde{Y}_k^{L,v}\}$
CFCA
$\mathbf{X}^v$
S
DISA
DISA
DISA
DISA
$\{\tilde{W}_k^v\}$
$1 \times 1 \times H^s \times W^s$
$\tilde{W}_1^v$
$\tilde{W}_2^v$
$\tilde{W}_{K-1}^v$
$\tilde{W}_K^v$
Mean
Mean
Mean
Mean
$\mathbf{L} \in \mathbb{R}^{K \times 256}$
$\mathbf{L}_{1,1}$ $\mathbf{L}_{1,2}$ ... $\mathbf{L}_{1,v+1}$ ... $\mathbf{L}_{1,256}$
$\mathbf{L}_{2,1}$ $\mathbf{L}_{2,2}$ ... $\mathbf{L}_{2,v+1}$ ... $\mathbf{L}_{2,256}$
$\mathbf{L}_{K-1,1}$ $\mathbf{L}_{K-1,2}$ ... $\mathbf{L}_{K-1,v+1}$ ... $\mathbf{L}_{K-1,256}$
$\mathbf{L}_{K,1}$ $\mathbf{L}_{K,2}$ ... $\mathbf{L}_{K,v+1}$ ... $\mathbf{L}_{K,256}$
K 1D LUTs for K rows
$\{\tilde{Y}_k^{L,v}\}: \{\tilde{Y}_k^L\}_p = v \quad \forall v \in \{0, 1, \ldots, 255\}$

Figure 2. **Top**: Our network structure $\mathcal{H}(\cdot)$ for unsupervised training. **Bottom**: Generation of K 1D LUTs. For each grayscale v, we feed the network with a group of YUV images with constant color v and set the LUT matrix by the predicted results.

and so on. [41, 42] proposed image-adaptive 3D LUTs for efficient single image enhancement and they all need a network weight predictor to fuse different 3D LUTs, which will be restricted for some platforms that require deep learning framework support. [43] train a deep super-resolution (SR) network with a restricted receptive field and then cache the output values of the learned SR network to the LUTs. Compared with [41, 42], MEFLUT and SR-LUT [43] are offline LUTs. In addition, MEFLUT is also essentially different from SR-LUT [43] in the form of generating LUT and training strategy.

3. Algorithm

Our method is divided into two stages, where Stage 1 is training a network $\mathcal{H}(\cdot)$ to estimate weight maps $\{W_k\}$ for a given set of K YUV input images $\{I_k^L\}, k = 1, \ldots, K$, and Stage 2 is generating K 1D lookup tables (LUTs) of size 256, composing a $K \times 256$ matrix $\mathbf{L} \in \mathbb{R}^{K \times 256}$, to achieve fast calculation of $\{W_k\}$ by simply querying $\mathbf{L}$ during deployment. With $\{W_k\}$ obtained, the output image I^H is obtained by alpha blending $\{I_k^L\}$ and $\{W_k\}$.

3.1. Network Structure

Our network $\mathcal{H}(\cdot)$ is built upon DNN with two newly proposed modules, which aim at inter-frame feature fusion and intra-frame weight prediction by attention mechanism in various dimensions, respectively. It takes K YUV images $\{I_k^L\}$ of size $H \times W$ as input, predicts K weight maps $\{W_k\}$ of the same size as intermediate results, and finally produces the output image I^H by alpha blending $\{I_k^L\}$ using $\{W_k\}$. Note that our network only involves computation of Y channel as it is sufficient to predict the weight maps as a common experience, while UV channels are separately merged using a simple weighted summation as in [5].

3.1.1 Convolution with frame and channel attention

The Y channel of the input frames $\{I_k^L\}$, i.e. $\{Y_k^L\} \in \mathbb{R}^{1 \times H \times W}$ are first bilinearly downsampled at rate s to obtain $\{\tilde{Y}_k^L\}$ with $\tilde{Y}_k^L \in \mathbb{R}^{1 \times H^s \times W^s}$. Then they are fed into a layer composed of convolution with frame and channel attention [45] (CFCA) to obtain a feature map $\mathbf{X}$. Specifically, we first separately apply conv2d to each $\tilde{Y}_k^L$ and concatenate the results together to get a 4D tensor $\mathbf{Y} \in \mathbb{R}^{K \times C \times H^s \times W^s}$, and then we successively apply attention mechanism in the frame and channel dimension as,
1) Channel attention:

$$\begin{aligned} p_{k,c}^{\mathbf{C}} &= \frac{1}{H^s \times W^s} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{Y}_{k,c}(i,j), \\ \mathbf{Y}_k^{\mathbf{C}} &= \mathbf{Y}_k \odot \sigma\big(\mathbf{W}_2^{\mathbf{C}} \cdot \delta(\mathbf{W}_1^{\mathbf{C}} \cdot p_k^{\mathbf{C}})\big), \\ \mathbf{Y}^{\mathbf{C}} &= \{\mathbf{Y}_k^{\mathbf{C}}\}, \quad k = 1, \ldots, K. \end{aligned} \tag{1}$$

2) Frame attention:

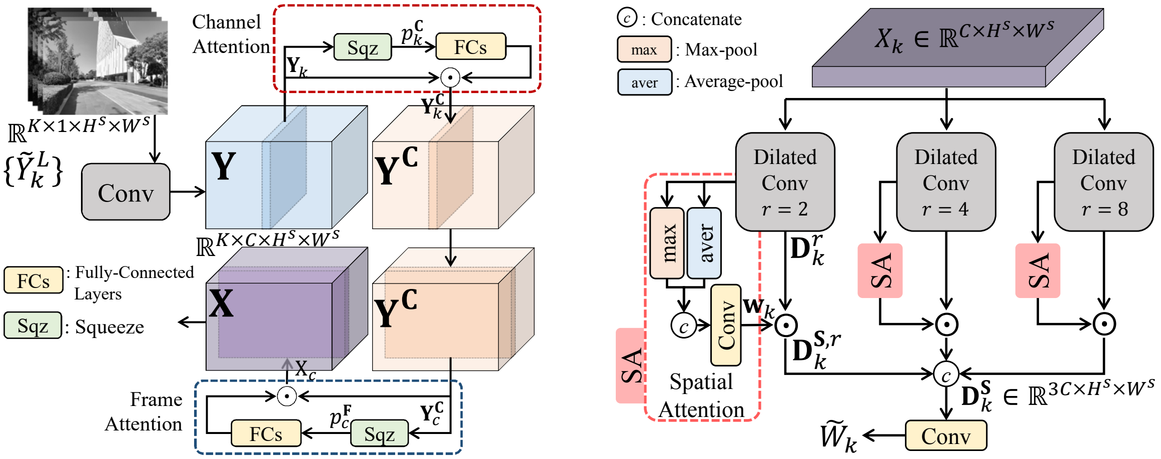


Figure 3. **Left**: Convolution with frame and channel attention (CFCA). **Right**: Dilated inception with spatial attention (DISA).

$$
\begin{aligned}
p^{\mathbf{F}}_{k,c} &= \frac{1}{H^s \times W^s}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{Y}^{\mathbf{C}}_{k,c}(i,j),\\
\mathbf{X}_c &= \mathbf{Y}^{\mathbf{C}}_c \odot \sigma\big(\mathbf{W}^{\mathbf{F}}_2\cdot\delta(\mathbf{W}^{\mathbf{F}}_1\cdot p^{\mathbf{F}}_c)\big),\\
\mathbf{X} &= \{\mathbf{X}_c\}, \quad c=1,...,C,
\end{aligned}
\tag{2}
$$

where $p^{\mathbf{C}}_k \in \mathbb{R}^{C\times 1\times 1}$, $p^{\mathbf{F}}_c \in \mathbb{R}^{K\times 1\times 1}$ are vectors composed of scalars $p^{\mathbf{C}}_{k,c}$ and $p^{\mathbf{F}}_{k,c}$, respectively. $\mathbf{W}^{\mathbf{C}}_1, \mathbf{W}^{\mathbf{C}}_2, \mathbf{W}^{\mathbf{F}}_1, \mathbf{W}^{\mathbf{F}}_2$ are linear weights in two attention modules, and δ, σ are ReLU and sigmoid activation. $\odot$ is element-wise product with broadcasting automatically.

After the above steps, the two attention modules fuse the feature in frame and channel dimensions separately, so that the produced feature map $\mathbf{X}$ accumulates rich context to predict the weight maps $\{\widetilde{W}_k\}$ in the following steps. As seen in Fig. 4, by comparing (e) and (c), or (d) and (b), we observe that with CFCA enabled, the weight map of the sky area (in green box) in EV-2 is highlighted, causing the final output image not over-saturated. This phenomenon proves the attention focuses more on EV-2 with CFCA involved. We illustrate CFCA structure in Fig. 3(**Left**).

3.1.2 Dilated inception with spatial attention

We further split $\mathbf{X}$ in frame dimension into $X_1, X_2, ..., X_K$ ($X_k \in \mathbb{R}^{C\times H^s\times W^s}$), and for each X_k, we feed it into a dilated inception layer with spatial attention (DISA). This layer consists of 3 branches of dilated convolution in rate $r = 2, 4, 8$ respectively, each of which produces a feature map with spatial attention [46] being applied to. Mathematically, the process is

$$
\begin{aligned}
\mathbf{D}^r_k &= \mathcal{D}(X_k, r) \in \mathbb{R}^{C\times H^s\times W^s}, \qquad r\in\{2,4,8\},\\
\mathbf{w}_k &= \sigma\left(\left[\frac{1}{C}\sum_{c=1}^{C}\mathbf{D}^r_k(c),\ \max_c \mathbf{D}^r_k(c)\right]\circledast \mathbf{W}^{\mathcal{D}}\right)\\
\mathbf{D}^{\mathbf{S},r}_k &= \mathbf{D}^r_k\odot\mathbf{w}_k,
\end{aligned}
\tag{3}
$$

where $\mathcal{D}(\cdot, r)$ is dilated conv2d with rate r, $[\cdot,\cdot]$ is concatenation and $\circledast$ is conv2d operator. With $\mathbf{D}^{\mathbf{S},r}_k$ in 3 branches obtained, we concatenate them together and apply another convolution so as to produce the final weight map $\widetilde{W}_k$. With DISA involved, the weights for pixels within a frame is spatially optimized so that details can be well preserved and artifacts are suppressed. For example in Fig. 4, we observe that the weights in red boxes become smoother from (e) to (d) or from (c) to (b), where (d)(b) additionally involve DISA based on (e)(c).

Having completed the above steps, we obtain a multi-scale spatial attention from different dilated rates, so that the produced feature map obtain finer spatial structural information to predict the weight maps $\{\widetilde{W}_k\}$ in the following steps. The detailed structure of DISA is illustrated in Fig. 3(**Right**). C is set to 24 in all modules, and we make sure $\min\{H^s, W^s\} = 128$ regardless of the input image size by controlling s.

We conduct an ablation study to demonstrate the effectiveness of the two attention modules, where with either module enabled, all metrics become higher.

3.1.3 Unsupervised learning of the MEFLUT

With $\{\widetilde{W}_k\}$ obtained, we further learn a high-quality detailed image in an unsupervised manner.

Guided filtering for upsampling (GFU). We first apply guided filtering for upsampling (GFU) [7] to resize $\{\widetilde{W}_k\}$ back to $\{W_k\}$ with its original resolution $H \times W$. This process also relies on $\{\tilde{Y}^L_k\}$ and $\{Y^L_k\}$ which provide guiding information to restore visual pleasing weight maps. These weight maps further alpha blends the Y channels $\{Y^L_k\}$ to achieve Y^H, i.e.

$$
Y^H = \sum_{k=1}^{K} W_k \cdot Y^L_k. \tag{4}
$$

With GFU involved, higher quality output can be achieved than using bilinear upsampling.

Loss function. As normally ground-truth detailed images are rarely available, we apply the MEF-SSIM loss in [7] to train our network in an unsupervised way, i.e.

$$
\min_{\theta} \mathcal{L}_{\text{MEF-SSIM}}(\{Y^L_k\}, Y^H), \tag{5}
$$

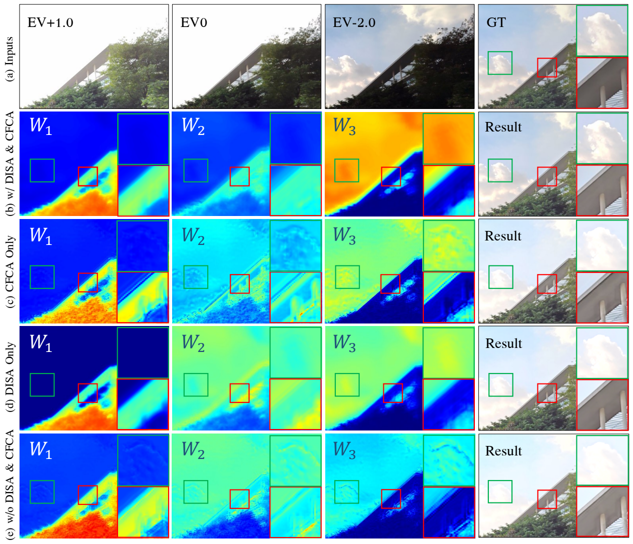


Figure 4. Visualization of weight maps produced by various cases. (a) Inputs and the GT target. From (b) to (e), weight maps of the 3 frames (column 1 to 3) and the final fusion image (column 4), for various cases. For weight maps, color map from deep blue to red represents value range $[0, 1]$.

where θ represents all the parameters in our network.

U/V channels' merging. For U/V channels, we adopt the weighted sum method in [5] to merge various frames into one, i.e.

$$P^H = \frac{\sum_{k=1}^{K} ||P_k - \tau||_1^1 \cdot P_k}{\sum_{k=1}^{K} ||P_k - \tau||_1^1}, \quad P \in \{U, V\}, \tag{6}$$

where $\tau = 128$. With all channels obtained, the output I^H is composed.

3.2. LUT Generation

Unlike other image algorithms which are sufficiently deployed in PC or cloud platform, MEF commonly works in mobile platform and it requires extreme efficiency. To achieve it, we further involve a LUT matrix $\mathbf{L} \in \mathbb{R}^{K \times 256}$ to speed up the inference procedure, with its k-th row corresponding to a 1D LUT for the k-th exposure image. To build up $\mathbf{L}$, we feed the trained network $\mathcal{H}(\cdot)$ with 256 groups of constant grayscale images, with grayscale varying from 0 to 255. For example, for the $(v+1)$-th image group, we set $\{\widetilde{Y}_k^L\}(i,j) = v, \forall v \in \{0, 1, ..., 255\}$ where (i,j) is a pixel location. This constant grayscale image group $\{\widetilde{Y}_k^{L,v}\}$ produces weight maps $\{\widetilde{W}_k^v\}$ which are further frame-wisely averaged into K scalars $\{\omega_k^v\}$, i.e. $\omega_k^v = \text{Mean}(\widetilde{W}_k^v)$. We set LUT matrix $\mathbf{L}$ by

$$\mathbf{L}(k, v+1) = \omega_k^v, \quad k = 1, ..., K; v = 0, ..., 255, \tag{7}$$

so that all the entries are filled after 256 groups of images are fed. During deployment, given a set of K input images $\{Y_k^L\}$, instead of feeding the network $\mathcal{H}$ again, we directly query $\mathbf{L}$ for each pixel location (i,j) where $v^* = Y_k^L(i,j)$, and set the weight map value by

$$\widetilde{W}_k(i,j) = \mathbf{L}(k, v^* + 1). \tag{8}$$

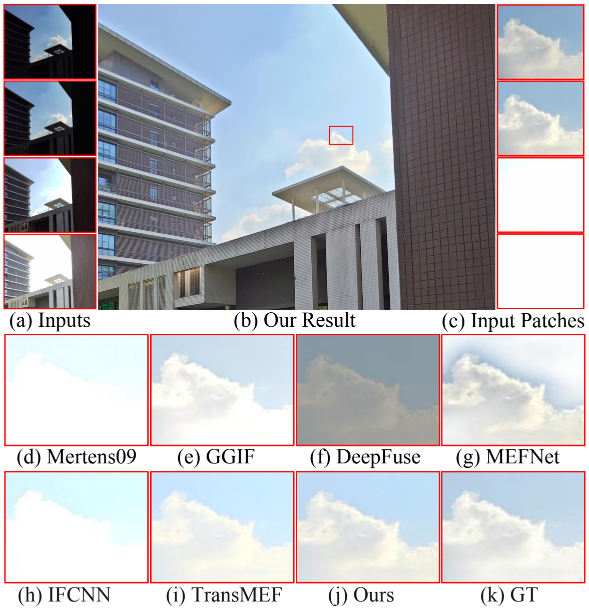


Figure 5. Qualitative comparison of (j) our method with (d) Mertens09, (e) GGIF, (f) DeepFuse, (g) MEFNet, (h) IFCNN and (i) TransMEF on our test set. (a) is inputs.

For a 4K image, this strategy makes the time cost around 3ms which leaves enough room for mobile platform deployment, and ensures our method outperforms the others in efficiency. For example, inference with our network directly costs 16ms in NVIDIA 2080Ti GPU. Detailed comparison data can be found in Fig. 7 and Tab. 2. We also visualize a LUT in Fig. 8 for clearer illustration.

4. Data Preparation

Considering there are few existing datasets of multi-exposure image sequences from mobile phones, and the provided image sequences as in [47, 39, 40] are very limited in quantity and diversity, we create a new dataset that contains higher number of multi-exposure image sequences and covers more diverse scenes captured by mobile phones.

Data collection. We collect the data mainly in static scenes using 6 different commonly used brands of mobile phones. The scenes are ensured diverse and representative, which cover a broad of scenarios, subjects, and lighting conditions. More importantly, the collected images cover most of the exposure levels in our daily life. For each sequence, we use a tripod to ensure the frames are well-aligned. The exposure levels are manually set, and Exposure Values (EVs) of our sample sequences setting range from -4.0 to $+2.0$ with 0.5 as a step. We select the exposure number 6 ($K = 6$) for each scene based on the characteristics of different brands of mobile phones. After collecting the source sequences, screening is further conducted to select desirable sequences for GT generation. As a result, a total of 960 static but diverse sequences are filtered out.

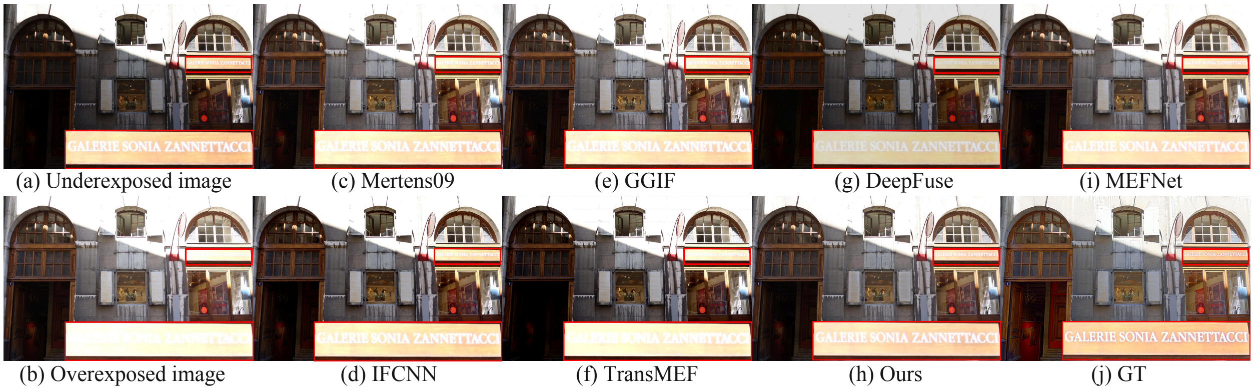


Figure 6. Qualitative comparison of (h) our method with (c) Mertens09, (e) GGIF, (g) DeepFuse, (i) MEFNet, (d) IFCNN and (f) TransMEF on SICE dataset. Input sequences with different exposures are shown in (a) and (b).

GT generation. We further use a hybrid method to generate the GT. Specifically, 14 existing algorithms [17, 48, 49, 30, 20, 50, 21, 22, 24, 51, 52, 53, 54, 7] were first used to predict results for each sequence. Then we invited 20 volunteers to compare the results of 14 algorithms and vote for an image as the GT for each sequence. We also invited an image quality tuning engineer to further manually tune the tone-mapping operators for the fairly voted results to generate the high-quality GT. The average tuning time for each image is 10 minutes, and each volunteer spent over 60 minutes for the entire voting. Due to the huge workload, we tuned and obtained high-quality GT on the 155 samples of test set only to evaluate the results by unsupervised learning, which costs totally $2450/60 = 40.8$ man-hours.

5. Experimental Results

5.1. Implementation Details

Dataset. We conduct experiments on our dataset, and another public static dataset SICE [39] is involved to keep the comparison fair enough. In our dataset, a total of 960 samples, of which 805 samples are used for training and 155 samples are used for evaluation. For SICE, we just utilized its first part of 360 index-available sequences with more classic pictures the authors obtained from a widely used dataset HDREYE [40]. Following the setting of SICE, 302 samples for training, remaining 58 samples for evaluation.

Evaluation. We use PSNR, SSIM, and another unsupervised metric Q$_C$ [55] as the main evaluation metric for all the methods we compare, besides the training loss MEF-SSIM [56]. Also, considering most of the involved methods are unsupervised ones, causing a large difference in brightness that may exist in their results, we calculate PSNR after an average brightness is subtracted from the given image for fair comparison.

Training. We fix the downsampling rate s to 4 so that the input training image are fixed size of 512^2. We use the ADAM optimizer [57] with momentum terms (0.9, 0.999), and the learning rate is set to $1e^{-4}$. The total training epoch is 100. Finally, we evaluate our method at full resolution during testing.

5.2. Comparison with Existing Methods

We compare our method with 8 previous MEF methods on our test set and SICE test set, including Mertens09 [17], Li13 [20], GGIF [22], Li20 [24], DeepFuse [5], MEFNet [7], IFCNN [8], and TransMEF [14]. The results of each method are generated by the implementations from the original authors with default settings.

Qualitative comparison. Fig. 5 shows the visual comparisons of fused images generated through other methods on our test set. As seen, the results of the other methods commonly fail to recover the details of saturated regions in the sky. In contrast, our method hallucinates more accurate contents in over saturated areas, causing the sharp edge of the cloud well preserved. Fig. 6 shows the visual comparisons on SICE dataset. As seen, the red box highlights that other methods produce a color cast in the billboard area without recovering the text details, while in contrast, our method produces a natural appearance with faithful detail and color restoration.

Quantitative comparison. As shown in Tab. 1(a), we conduct experiments on three sub-datasets with exposure numbers set to 2, 3, and 4, we select the exposures we need from each sample at multiple fixed EVs intervals. As seen from the table, our method consistently achieves the best results over the others in various exposure numbers. We also evaluate all methods on SICE. Here we conduct experiments with the exposure number being set to 2 (EV-1, EV+1), this setting follows the index-available sequences provided by SICE. As seen from the Tab. 1(b), our method achieves similar results, verifying the generalization of our method.

Running time. We conduct a speed comparison of our method with the 8 MEF methods on our test set, in various platforms including CPU, GPU and mobile CPU, as shown in Tab. 2. There, CPU is an Intel i7-10510U 1.8GHz, GPU is an Nvidia GeForce RTX 2080Ti and the mobile

Exposure Number (K)	Metrics	Mertens09	Li13	GGIF	Li20	DeepFuse	MEFNet	IFCNN	TransMEF	Ours
$K = 2$	PSNR↑	26.886	19.475	27.172	25.997	23.087	28.763	29.085	29.285	29.443
	SSIM↑	0.8347	0.8326	0.9153	0.8765	0.7558	0.9468	0.9373	0.9568	0.9587
	Q_C ↑	0.5531	0.5166	0.5330	0.5245	0.4603	0.6596	0.6322	0.6796	0.6860
	MEF-SSIM↑	0.9410	0.8797	0.9638	0.9273	0.7407	0.9633	0.9683	0.9663	0.9773
$K = 3$	PSNR↑	27.328	20.922	29.437	25.396	-	29.793	30.291	-	31.387
	SSIM↑	0.8927	0.8468	0.9208	0.8603	-	0.9569	0.9473	-	0.9590
	Q_C ↑	0.4779	0.4562	0.4735	0.4742	-	0.6054	0.5798	-	0.6075
	MEF-SSIM↑	0.9452	0.8955	0.9592	0.9387	-	0.9593	0.9612	-	0.9688
$K = 4$	PSNR↑	26.154	20.471	27.782	24.602	-	29.365	29.872	-	31.025
	SSIM↑	0.8834	0.8276	0.9087	0.8448	-	0.9513	0.9501	-	0.9548
	Q_C ↑	0.4379	0.4107	0.4359	0.4371	-	0.5493	0.5421	-	0.5571
	MEF-SSIM↑	0.9360	0.8897	0.9528	0.9096	-	0.9512	0.9543	-	0.9603

(a) Results on our dataset.

Metrics	Mertens09	Li13	GGIF	Li20	DeepFuse	MEFNet	IFCNN	TransMEF	Ours
PSNR↑	21.531	19.380	21.316	21.072	20.915	21.583	21.626	21.601	21.894
SSIM↑	0.7833	0.7732	0.7929	0.7743	0.6872	0.7763	0.7839	0.7965	0.8067
Q_C ↑	0.7759	0.7056	0.7709	0.7747	0.6406	0.7760	0.7789	0.7837	0.7934
MEF-SSIM↑	0.9589	0.9451	0.9692	0.9498	0.8590	0.9743	0.9756	0.9782	0.9845

(b) Results on SICE [39] dataset.

Table 1. The PSNR, SSIM, Q_C, MEF-SSIM of all methods on our test set (a) and SICE dataset (b). The best and 2nd best results are highlighted in red and in blue, respectively. '-' means not applicable, as DeepFuse and TransMEF can only take two images as input.

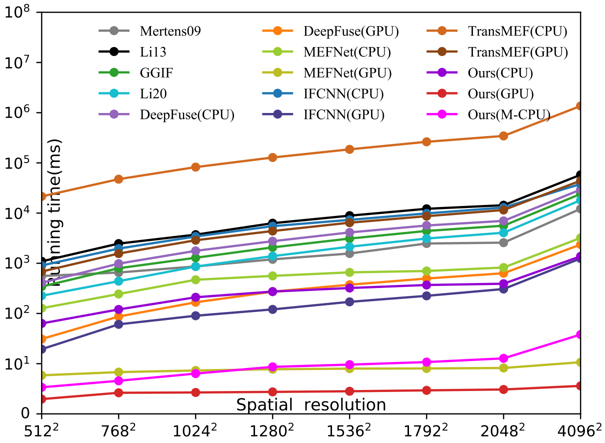


Figure 7. Running time in milliseconds (ms) on different spatial resolutions ($K = 2$).

	Methods	Exposure Number (K)		
		2	3	4
CPU	Mertens09	2575	3570	4624
	Li13	14336	17708	20479
	GGIF	5605	8268	10861
	Li20	4057	5973	7835
	DeepFuse	6991	-	-
	MEFNet	822(14833)	861(15097)	890(15641)
	IFCNN	12901	18123	23954
	TransMEF	344168	-	-
	Ours	391(54)	413(56)	435(60)
GPU	DeepFuse	634	-	-
	MEFNet	8.21(5.48)	8.65(5.92)	8.72(6.31)
	IFCNN	307	360	417
	TransMEF	11524	-	-
	Ours	3.06(0.53)	3.54(0.61)	3.89(0.65)
M-CPU	Ours	11.78	12.75	13.32

Table 2. Running time of MEF methods on different K settings in milliseconds (ms). '-' means not supported by DeepFuse and TransMEF for such a test. For values in '()', we only compare our method with MEFNet, they are calculated by disabling GFU so that the speed of LUT only can be directly measured. M-CPU denotes the mobile CPU. The best and 2nd best are marked in red and blue, respectively.

CPU is Qualcomm SM8250. Mertens09, Li13, GGIF and Li20 utilize CPU only, while DeepFuse, MEFNet, IFCNN and TransMEF exploit CPU and GPU, and ours exploit all three. We only compared the running time of DeepFuse and TransMEF on 2 exposures due to its strict input constraints.

When the resolution is fixed to 2048^2 (2K), our method run in 3.06ms in GPU, which is faster than MEFNet's speed (8.21ms) and DeepFuse's speed (634ms), and the speed of CPU is also significantly faster than that of DeepFuse's, MEFNet's and other methods such as Mertens09, Li13, GGIF, Li20, IFCNN and TransMEF, these methods show a lot of performance consumption. In mobile CPU, our method is able to achieve real-time performance. Compared with PC CPU, our method optimizes the performance of query and other operations for mobile CPU, and our method has no strict requirements on the running platform compared with deep learning methods such as Deepfuse, MEFNet, IFCNN and TransMEF.

We also conduct experiments on images with various resolutions and illustrate the performance in Fig. 7. As seen, our method takes less than 4ms to process image sequences with resolutions ranging from 512^2 to 4096^2 on the GPU. The network of MEFNet downsamples the full resolution image to a fixed 128^2, so their running time is relatively not much. Our method maintains the quality while achieving less running time, which is of 3.0× and 175.0× speed of MEFNet and DeepFuse, and 345.4× and 12226.0× speed of IFCNN and TransMEF in 4096^2 resolution (4K). In CPU, our method is of 21.3× and 42.0× speed of DeepFuse and Li13, and 27.6× and 989.6× speed of IFCNN and TransMEF. In mobile CPU, our method running less than 40ms for a 4K image sequence. We further conduct an experi-

CFCA	DISA	PSNR↑	SSIM↑	Q_C↑	MEF-SSIM↑
		29.648	0.9373	0.5617	0.9442
✓		30.570	0.9431	0.5943	0.9522
	✓	30.782	0.9482	0.5960	0.9598
✓	✓	**31.387**	**0.9590**	**0.6075**	**0.9688**

Table 3. Ablation study of the proposed CFCA and DISA modules ($K = 3$).

Methods	PSNR↑	SSIM↑	Q_C↑	MEF-SSIM↑
MEFNet	25.315	0.8924	0.5575	0.9064
Ours	**31.387**	**0.9590**	**0.6075**	**0.9688**

Table 4. The Effectiveness of our network to LUTs ($K = 3$).

ment by working on images in their original resolution by disabling the GFU module to verify the efficiency of our 1D LUTs only. The performance advantage of our luts is more obvious when GFU disabled.

Our method has been successfully shipped into millions of mobile phones in the market. This fact further demonstrates the practicality and robustness of our method.

5.3. Ablation Studies

Visualization of LUTs. Fig. 8 shows the weight value of the multi-exposure image in the range of 0 to 255 in 3 EVs. As seen, the curve of EV+ basically follows a monotonously decreasing pattern, while that of EV− follows a monotonously increasing one. Our 1D LUTs reflect the fusion trend of different exposure images.

The effectiveness of CFCA and DISA. We verify the module of CFCA and DISA respectively. Tab. 3 and Fig. 4 shows that when the module of CFCA and DISA are also adopted, all metrics and visual effects can achieve the best performance. The main reason can be concluded as that the dual attention mechanism of the CFCA strengthens the exchange of information between different exposures, our 1D LUTs can better take into account the difference in brightness between different exposures, and the DISA acts on each branch and focuses on spatial features under different receptive fields, the consistency of spatial information can be preserved as much as possible.

The effectiveness of our network to LUTs. In order to verify that the network we proposed is more effective for 1D LUTs' generation, we apply the same idea to MEFNet and used the generated 1D LUTs to perform the fusion task instead. As seen in Tab. 4, although MEFNet produces similar 1D LUTs to reconstruct the weight maps, its quality has a significant drop. It demonstrates that our network has better effectiveness for 1D LUTs' generation.

Moving scene. In our paper, we focus on MEF, similar to previous MEF works, such as DeepFuse, MEFNet, IFCNN and TransMEF, we all focus on static scenes. However, we also show how MEFLUT collaborates with other deghosting methods in practical applications. We show an example captured by a hand-held cellphone with dynamic objects in Fig. 9. We first align the input frames, then remove ghosts in the final image. As for solving the ghosts, we generate weight maps based on our 1D LUTs. The moving areas are detected and marked according to [58]. The marked pixels are then assigned zero weights to prevent ghost artifacts. Our method can collaborate well with the off-the-shelf deghosting methods for practical results.

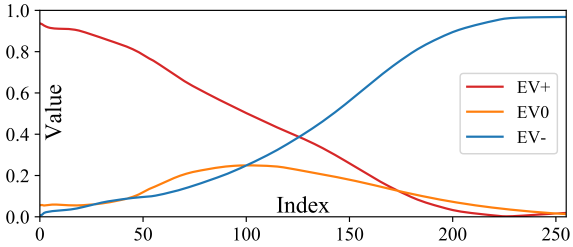
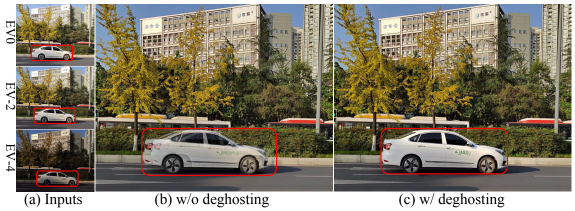


Figure 8. The visualization of the 1D LUTs ($K = 3$).



(a) Inputs (b) w/o deghosting (c) w/ deghosting

Figure 9. MEFLUT collaborates with a deghosting method.

Methods	PSNR↑	SSIM↑	Q_C↑	MEF-SSIM↑	Running time (ms)
Network	**32.125**	**0.9674**	**0.6168**	**0.9764**	12.37
LUTs	31.387	0.9590	0.6075	0.9688	**3.54**

Table 5. Comparison experiment of our LUTs with our network. The running platform is PC GPU ($K = 3$).

Trained network vs. LUTs. Although the LUTs in our method do not participate in the training. As shown in Tab. 5, we find that the quality of the results of the LUTs is comparable to the trained network, but the speed is several times faster. Our method has little effect in actual deployment, but using LUTs will get a large speed advantage with less quality loss.

6. Conclusion

We have proposed a new method to efficiently fuse a multi-exposure image sequence to produce a visually pleasing result. We learn one 1D LUT for each exposure, then all the pixels from different exposures can query 1D LUT of that exposure independently for high-quality and efficient fusion. Specifically, to learn these 1D LUTs, we involve frame, channel and spatial attention mechanism into the MEF task to achieve superior performance. We also release a new dataset composed of 960 multi-exposure image sequences collected from mobile phones of various brands and in diverse scenes. We have conducted comprehensive experiments to demonstrate the effectiveness of MEFLUT.

Acknowledgements This work is supported by Sichuan Science and Technology Program under grant No.2023NSFSC0462.

References

[1] Luca Bogoni. Extending dynamic range of monochrome and color images through fusion. pages 7–12, 2000. 1

[2] Qingsen Yan, Dong Gong, Qinfeng Shi, Anton van den Hengel, Chunhua Shen, Ian Reid, and Yanning Zhang. Attention-guided network for ghost-free high dynamic range imaging. In *Proc. CVPR*, pages 1751–1760, 2019. 1

[3] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *Pacific Conference on Computer Graphics and Applications*, pages 382–390, 2007. 1, 2

[4] Zhengguo Li, Zhe Wei, Changyun Wen, and Jinghong Zheng. Detail-enhanced multi-scale exposure fusion. *IEEE Trans. on Image Processing*, 26(3):1243–1252, 2017. 1

[5] K Ram Prabhakar, V Sai Srikar, and R Venkatesh Babu. Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs. In *Proc. CVPR*, pages 4714–4722, 2017. 1, 2, 3, 5, 6

[6] Hui Li and Lei Zhang. Multi-exposure fusion with cnn features. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1723–1727, 2018. 1

[7] Kede Ma, Zhengfang Duanmu, Hanwei Zhu, Yuming Fang, and Zhou Wang. Deep guided learning for fast multi-exposure image fusion. *IEEE Trans. on Image Processing*, 29:2808–2819, 2019. 1, 2, 4, 6

[8] Yu Zhang, Yu Liu, Peng Sun, Han Yan, Xiaolin Zhao, and Li Zhang. Ifcnn: A general image fusion framework based on convolutional neural network. *Information Fusion*, 54:99–118, 2020. 1, 2, 6

[9] Han Xu, Jiayi Ma, Junjun Jiang, Xiaojie Guo, and Haibin Ling. U2fusion: A unified unsupervised image fusion network. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44(1):502–518, 2020. 1, 2

[10] Han Xu, Jiayi Ma, and Xiao-Ping Zhang. Mef-gan: Multi-exposure image fusion via generative adversarial networks. *IEEE Trans. on Image Processing*, 29:7203–7216, 2020. 1

[11] Hao Zhang, Han Xu, Yang Xiao, Xiaojie Guo, and Jiayi Ma. Rethinking the image fusion: A fast unified image fusion network based on proportional maintenance of gradient and intensity. In *Proc. AAAI*, volume 34, pages 12797–12804, 2020. 1

[12] Hyungjoo Jung, Youngjung Kim, Hyunsung Jang, Namkoo Ha, and Kwanghoon Sohn. Unsupervised deep image fusion with structure tensor representations. *IEEE Trans. on Image Processing*, 29:3845–3858, 2020. 1

[13] Han Xu, Jiayi Ma, Zhuliang Le, Junjun Jiang, and Xiaojie Guo. Fusiondn: A unified densely connected network for image fusion. In *Proc. AAAI*, volume 34, pages 12484–12491, 2020. 1, 2

[14] Linhao Qu, Shaolei Liu, Manning Wang, and Zhijian Song. Transmef: A transformer-based multi-exposure image fusion framework using self-supervised multi-task learning. In *Proc. AAAI*, volume 36, pages 2126–2134, 2022. 1, 2, 6

[15] Hao Zhang and Jiayi Ma. Iid-mef: A multi-exposure fusion network based on intrinsic image decomposition. *Information Fusion*, 95:326–340, 2023. 1, 2

[16] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proc. ACM SIGGRAPH*, pages 1–10. 2008. 1

[17] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. In *Computer graphics forum*, volume 28, pages 161–171, 2009. 1, 2, 6

[18] Miguel Granados, Boris Ajdin, Michael Wand, Christian Theobalt, Hans-Peter Seidel, and Hendrik PA Lensch. Optimal hdr reconstruction with linear digital cameras. In *Proc. CVPR*, pages 215–222, 2010. 1

[19] Zheng Guo Li, Jing Hong Zheng, and Susanto Rahardja. Detail-enhanced exposure fusion. *IEEE Transactions on Image Processing*, 21(11):4672–4676, 2012. 1

[20] Shutao Li, Xudong Kang, and Jianwen Hu. Image fusion with guided filtering. *IEEE Trans. on Image Processing*, 22(7):2864–2875, 2013. 1, 2, 6

[21] Kede Ma, Hui Li, Hongwei Yong, Zhou Wang, Deyu Meng, and Lei Zhang. Robust multi-exposure image fusion: a structural patch decomposition approach. *IEEE Trans. on Image Processing*, 26(5):2519–2532, 2017. 1, 6

[22] Fei Kou, Zhengguo Li, Changyun Wen, and Weihai Chen. Multi-scale exposure fusion via gradient domain guided image filtering. In *Proc. ICME*, pages 1105–1110, 2017. 1, 2, 6

[23] Kede Ma, Zhengfang Duanmu, Hojatollah Yeganeh, and Zhou Wang. Multi-exposure image fusion by optimizing a structural similarity index. *IEEE Trans. on Computational Imaging*, 4(1):60–72, 2017. 1, 2

[24] Hui Li, Kede Ma, Hongwei Yong, and Lei Zhang. Fast multi-scale structural patch decomposition for multi-exposure image fusion. *IEEE Trans. on Image Processing*, 29:5805–5816, 2020. 1, 2, 6

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Proc. NeurIPS*, 28:91–99, 2015. 1

[26] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015. 1

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 1

[29] A Ardeshir Goshtasby. Fusion of multi-exposure images. *Image and Vision Computing*, 23(6):611–618, 2005. 2

[30] Wei Zhang and Wai-Kuen Cham. Gradient-directed multi-exposure composition. *IEEE Trans. on Image Processing*, 21(4):2318–2323, 2011. 2, 6

[31] Shutao Li, Xudong Kang, Leyuan Fang, Jianwen Hu, and Haitao Yin. Pixel-level image fusion: A survey of the state of the art. *information Fusion*, 33:100–112, 2017. 2

[32] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM Trans. Graphics*, 36(6):177, 2017. 2

[33] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. 1987. 2

[34] Peter J Burt and Raymond J Kolczynski. Enhanced image capture through fusion. In *Proc. ICCV*, pages 173–182, 1993. 2

[35] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graphics*, 35(6):1–12, 2016. 2

[36] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Trans. Graphics*, 36(4):1–12, 2017. 2

[37] Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *Proc. ICCV*, pages 2497–2506, 2017. 2

[38] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2012. 2

[39] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Trans. on Image Processing*, 27(4):2049–2062, 2018. 2, 5, 6, 7

[40] Hiromi Nemoto, Pavel Korshunov, Philippe Hanhart, and Touradj Ebrahimi. Visual attention in ldr and hdr images. In *9th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, number CONF, 2015. 2, 5, 6

[41] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 44(4):2058–2073, 2020. 2, 3

[42] Tao Wang, Yong Li, Jingyang Peng, Yipeng Ma, Xian Wang, Fenglong Song, and Youliang Yan. Real-time image enhancer via learnable spatial-aware 3d lookup tables. In *Proc. ICCV*, pages 2471–2480, 2021. 2, 3

[43] Younghyun Jo and Seon Joo Kim. Practical single-image super-resolution using look-up table. In *Proc. CVPR*, pages 691–700, 2021. 2, 3

[44] Jiacheng Li, Chang Chen, Zhen Cheng, and Zhiwei Xiong. Mulut: Cooperating multiple look-up tables for efficient image super-resolution. In *Proc. ECCV*, pages 238–256, 2022. 2

[45] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proc. CVPR*, pages 7132–7141, 2018. 3

[46] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proc. ECCV*, pages 3–19, 2018. 4

[47] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. *ACM Trans. Graphics*, 22(3):319–325, 2003. 5

[48] Shanmuganathan Raman and Subhasis Chaudhuri. Bilateral filter based compositing for variable exposure photography. In *Eurographics*, pages 1–4, 2009. 6

[49] Rui Shen, Irene Cheng, Jianbo Shi, and Anup Basu. Generalized random walks for fusion of multi-exposure images. *IEEE Trans. on Image Processing*, 20(12):3634–3646, 2011. 6

[50] Jianbing Shen, Ying Zhao, Shuicheng Yan, Xuelong Li, et al. Exposure fusion using boosting laplacian pyramid. *IEEE Trans. Cybern.*, 44(9):1579–1590, 2014. 6

[51] Pradeep Sen, Nima Khademi Kalantari, Maziar Yaesoubi, Soheil Darabi, Dan B Goldman, and Eli Shechtman. Robust patch-based hdr reconstruction of dynamic scenes. *ACM Trans. Graphics*, 31(6):203–1, 2012. 6

[52] Jun Hu, Orazio Gallo, Kari Pulli, and Xiaobai Sun. Hdr deghosting: How to deal with saturation? In *Proc. CVPR*, pages 1163–1170, 2013. 6

[53] Neil DB Bruce. Expoblend: Information preserving exposure blending based on normalized log-domain entropy. *Computers & Graphics*, 39:12–23, 2014. 6

[54] Tae-Hyun Oh, Joon-Young Lee, Yu-Wing Tai, and In So Kweon. Robust high dynamic range imaging by rank minimization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 37(6):1219–1232, 2014. 6

[55] Nedeljko Cvejic, Artur Loza, David Bull, and Nishan Canagarajah. A similarity metric for assessment of image fusion algorithms. *International journal of signal processing*, 2(3):178–182, 2005. 6

[56] Kede Ma, Kai Zeng, and Zhou Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE Trans. on Image Processing*, 24(11):3345–3356, 2015. 6

[57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014. 6

[58] Benkang Zhang, Qin Liu, and Takeshi Ikenaga. Ghost-free high dynamic range imaging via moving objects detection and extension. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 459–462, 2015. 8