博 士 学 位 論 文

論文題目　Deep Learning for High-Fidelity Imaging beyond the Limits of Imaging Devices

提　出　者　　東北大学大学院情報科学研究科

システム情報科学　　専　攻

氏　名　　Qian YE

TOHOKU UNIVERSITY
Graduate School of Information Sciences

Deep Learning for High-Fidelity Imaging beyond the Limits of Imaging Devices
(デバイスの限界を超えた正確な撮像を可能にする深層学習)

A dissertation submitted for the degree of Doctor of Philosophy (Information Sciences)
Department of System Information Sciences

by

Qian YE

July 8, 2022

Deep Learning for High-Fidelity Imaging beyond the Limits of Imaging Devices

Qian YE

Abstract

Digital imaging devices have become more and more important in our daily life. However, due to environmental conditions and the limitations of imaging devices, the captured digital images are generally not desirable in visual quality. For example, real-world scenes have wide illumination ranges that exceed standard camera sensors' dynamic range; the standard cameras can only produce low dynamic range (LDR) images with under-exposure and over-exposure regions where the detailed information is missing. Furthermore, when the depth range of the scene is relatively large, but the camera's depth of field (DoF) is limited, the captured image can also be blurry due to defocus. Thus, there are increasing demands to improve the image quality from the limited devices. However, it is expensive to use a better camera to achieve this goal. A common and cost-effective approach is to employ image enhancement methods to improve the image quality from limited devices. Due to the advancement of deep learning, deep Convolutional Neural Network (CNN) has achieved impressive success in computer vision tasks (e.g., image classification, segmentation, detection, etc.) and have gradually become the dominant method in image enhancement tasks. In this dissertation, we consider improving the image quality from devices with the above limitations, aiming at solving the problem above in real-world applications. We focus on designing effective architectures to address these problems.

In Chapter 2, we consider the problem of generating a high dynamic range (HDR) image of a scene from its LDR images in a feature fusion manner. Recent studies employ deep learning and solve the problem end-to-end, leading to significant performance improvements. However, it is still hard to generate a good quality image from LDR images of a dynamic scene captured by a hand-held camera, e.g., occlusion due to the significant motion of foreground objects, causing ghosting artifacts. The key to success relies on how well we can fuse the input images in their feature space, where we wish to remove the factors leading to low-quality image generation while performing the fundamental computations for HDR image generation, e.g., selecting the best-exposed image/region. We propose a novel method that can better fuse the features based on two ideas. One is multi-step feature fusion; our network gradually

fuses the features in a stack of blocks having the same structure. The other is the design of the component block that effectively performs two operations essential to the problem, i.e., comparing and selecting appropriate images/regions. Experimental results show that the proposed method outperforms the previous state-of-the-art methods on the standard benchmark tests.

In Chapter 3, we further consider the HDR image reconstruction problem in an alignment-before-merging manner. Large object motion and occlusions in the LDR images often lead to visible artifacts using existing methods. To address this problem, we propose a deep network that tries to learn multi-scale feature flow guided by the regularized loss. It first extracts multi-scale features and then aligns features from non-reference images. After alignment, we use residual channel attention blocks to merge the features from different images. Extensive qualitative and quantitative comparisons show that our approach achieves state-of-the-art performance and produces excellent results where color artifacts and geometric distortions are significantly reduced.

In Chapter 4, we consider the problem in defocus image deblurring. Previous classical methods follow two-step approaches, i.e., first defocus map estimation and then the non-blind deblurring. In the era of deep learning, some researchers have tried to address these two problems by CNN. However, the simple concatenation of defocus map, which represents the blur level, leads to suboptimal performance. Considering the spatial variant property of the defocus blur and the blur level indicated in the defocus map, we employ the defocus map as conditional guidance to adjust the features from the input blurring images instead of simple concatenation. Then we propose a simple but effective network with spatial modulation based on the defocus map. To achieve this, we design a network consisting of three sub-networks, including the defocus map estimation network, a condition network that encodes the defocus map into condition features, and the defocus deblurring network that performs spatially dynamic modulation based on the condition features. Moreover, the spatially dynamic modulation is based on an affine transform function to adjust the features from the input blurry images. Experimental results show that our method can achieve better quantitative and qualitative evaluation performance than the existing state-of-the-art methods on the commonly used public test datasets.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital imaging devices such as smartphone cameras and digital cameras have become more and more important in our daily life. For example, due to the convenient usage and low price of the smartphone cameras, people are likely to use smartphone cameras to record the moments in their daily life. And also, people use surveillance cameras to guarantee the security of society. Therefore, there are increasing demands to improve the image quality from the limited devices. However, due to the constraints of environmental conditions and limited imaging devices, the captured digital images are generally not desirable in visual quality. For example, the real-world scenes have wide illumination ranges that exceeds the dynamic range of standard camera sensors; the standard cameras can only produce low dynamic range (LDR) images that have under-exposure and over-exposure regions where the detailed information is missing. And when the depth range of the scene is relatively large, but the depth of field (DoF) of the camera is limited, the captured image can also be blurry due to defocus. It is expensive to overcome these problems by using a better camera that has a larger depth of field or bit-depth. Thus, a common and more cost-effective approach is to employ image enhancement methods to improve the quality of the imaging from limited devices. It can effectively improve the image's visual quality without increasing too much imaging hardware cost.

Recently, deep Convolutional Neural Networks (CNNs) have achieved impressed success in computer vision tasks. The most successful application of CNNs in computer vision is image classification, i.e., classifying an image into its class (e.g., dog

or cat). In 2015, CNNs with residual connections [6] achieved humans' level accuracy on a large scale dataset [7]; CNNs with densely connected layers [8] further improved the performance on the same task. Building on the success of image classification, CNNs also achieve successful application on many other tasks, like object detection and segmentation [9–11], image generation [12] and so on, showing their great capacity to address various problems in computer vision. So in this dissertation, we utilize CNN-based high dynamic range imaging (HDRI) and defocus deblurring methods to improve the quality of the imaging from the limited devices.

In this chapter, we give a brief introduction on HDRI 1.1, image defocus deblurring 1.2. To encourage more understanding, we include the preliminary knowledge related to our study in Sec. 1.3. Lastly, in this chapter, the outline of each individual chapter is described in Sec. 1.4.

1.1 High Dynamic Range Imaging

High dynamic range imaging is an important task in computational photography and image processing which aims to recover the high dynamic range image from the low dynamic range images. The dynamic range is defined as the ratio of the highest to the lowest luminance in a scene. The dynamic range of a real-world scene is extremely high, which is approximately 14 orders of magnitude [13,14]. For example, the typical challenging scene for most consumers and even professional digital cameras is a scene with both dark/indoor and bright/outdoor scenes. Due to the limited range of the devices, in order to make scenes in the dark visible, the camera needs longer exposure times, but this will make the bright regions saturated or over-exposure. On the other hand, using shorter exposure times to capture bright area details will lose the information in the darker areas.

Accurate recovering the high dynamic range image for natural scenes is a challenging task [15]. A direct way to achieve HDR image is to use a specialized camera. However, this strategy is not widely used since such a specialized camera is expensive for ordinary usage. Another strategy is to reconstruct the HDR image based single/multiple LDR images captured by the limited sensors.

Generally, the LDR camera needs to set the appropriate exposure time (Δt^j) and

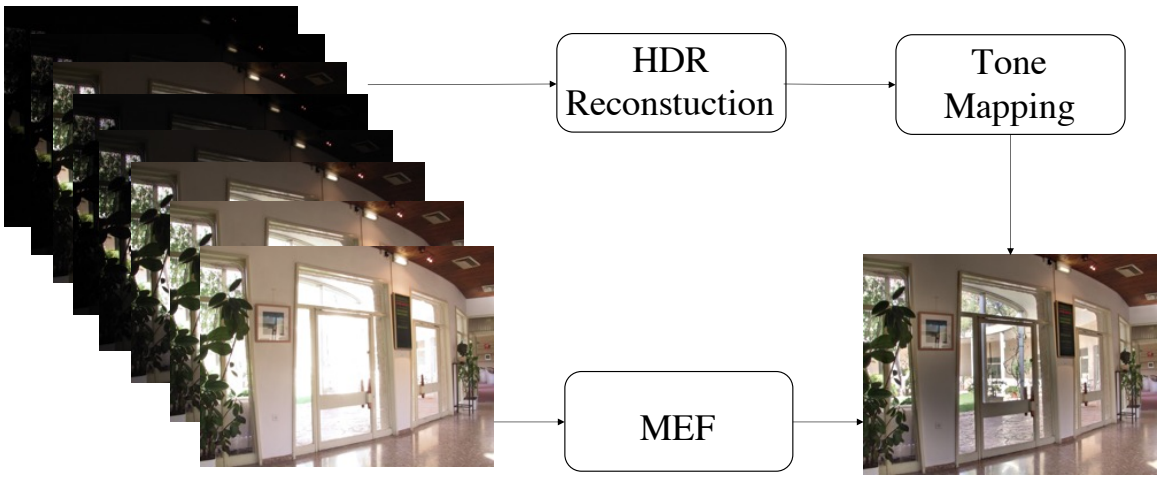


Figure 1.1: High dynamic range imaging based on multi-exposure images.

relies on the camera response function (CRF) f_{CRF} to map the irradiance (E) of the natural scene captured by the lens to the LDR image (x) [14]:

$$x_i^j = f_{CRF}(E_i^j \Delta t^j) \tag{1.1}$$

where E_i^j and x_i^j are the irradiance and pixel value at pixel location (i) in the j-th exposure image with exposure time Δt_j, respectively. For an image sequence, there are a set of multi-exposure images $X = x^1, ..., x^j, ..., x^J$, where J is the number of LDR images.

The simplest method for HDR reconstruction is to learn the inverse mapping of the CRF from a single-exposure LDR image. However, since only a single exposure image contains limited information for reconstruction, learning the HDR content from it is an ill-posed problem [14, 16]. Therefore, it is difficult to reconstruct the HDR image using a single exposure image, and the quality of the HDR image is inferior to that of using multi-exposure LDR images.

Another strategy for HDRI is to utilize a sequence of different exposure image. As shown in 1.1, there are two kinds of methods that utilize a sequence of different exposure image for HDRI: multi-exposure fusion methods [17, 18] in image domain and HDR reconstruction [1, 5, 19] in radiance domain and then tone mapping to LDR image for visualization.

As shown in Figure 1.2, multi-exposure fusion methods [17, 18] provide a simple

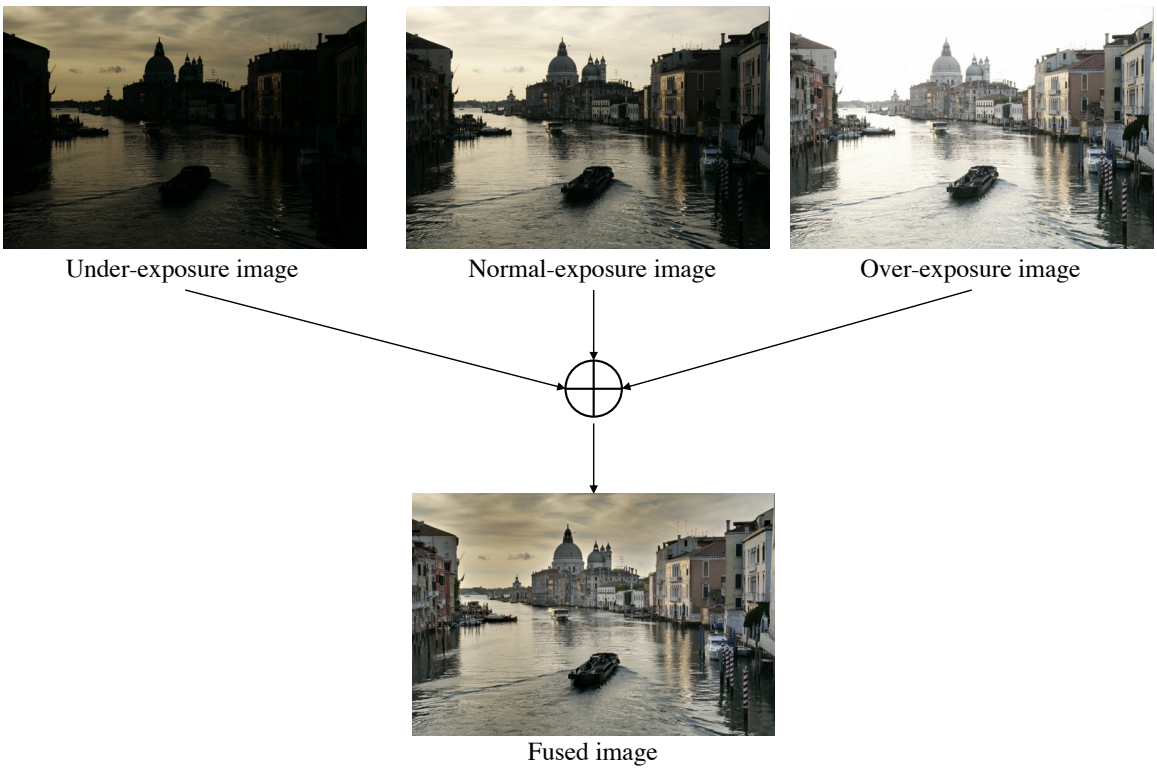


Figure 1.2: Multi-exposure image fusion for a static scene.

approach to achieve high dynamic range image based a fusion operation into the LDR images with different exposure times. It performs fusion in the 8-bit pixel value domain and has been widely used in mobile devices for HDR imaging [20].

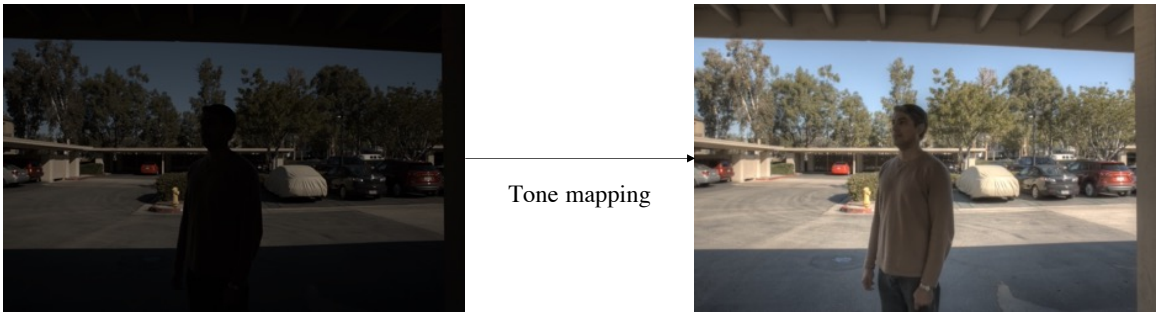
The fusion operation can be described as followed,

$$x_i = \sum_{j=1}^{J} W_i^j x_i^j \tag{1.2}$$

where W_i^j and x_i^j are the weights and pixel value at pixel location (i) in the j-th exposure image, respectively; x_i is the fused image.

Multi-exposure fusion methods are specially designed only for static scenes and have achieved good performance on the static scenes [17, 21, 22]. As shown in Figure 1.4, it is obvious that there would be ghosting artifacts since there are moving objects. However, in the real situation, there are always moving objects or a shaking camera, resulting in ghosting artifacts in the dynamic scene. In order to address this kind of ghosting artifact, the source exposure images are globally registered via registration

operators such as SIFT [23], Harris [24], SURF [25]. With global alignment, the multi-exposure fusion methods still perform on the pixel value domain, which limits the performance of HDR imaging in the dynamic scenes.



(a) high-bit image (b) 8-bit image

Figure 1.3: An example for tone mapping.

In recent years, deep learning-based methods have achieved unprecedented success in high dynamic range imaging and gradually become a dominant approach in this task. Unlike the multi-exposure fusion methods fused on the pixel value domain, the deep learning based approach to reconstruct HDR content in radiance domain can directly learn a highly non-linear and complicated mapping function from the given multi-exposure images to the HDR images. After achieving the HDR image, a tone mapping function needs to be applied for visualization since standard display devices like LCD cannot directly show the HDR image. Thus, the tone mapping function is used to compress the dynamic range of HDR images for effective visualization as shown in Figure 1.3. To this end, our goal is to design effective network structures on deep learning to achieve better performance on HDRI.

Under-exposure image Normal-exposure image Over-exposure image

Figure 1.4: A dynamic scene with multi-exposure.

1.2 Defocus Image Deblurring

Defocus blur is inevitable when the rays from a scene not lying on the focal plane of the camera converge to a region rather than a point on the image plane, as shown in 1.5, and the region is called the circle of confusion (COC) [26]. This phenomenon is caused by the limitations of the hardware. The exposure of an image is governed by two parameters: shutter speed and aperture size. Shutter speed indicates the time in which the sensor is exposed to light, while the aperture controls the amount of light traveling through the lens and falling on the image sensor. The camera can achieve the same exposure by fixing one of them while adjusting the other one. For example, when a camera is set into aperture-priority mode, the aperture is fixed while the shutter speed is adjusted to control how long the sensor is exposed to light. But the problem with the slow shutter speed is that it results in the motion blur if there is a moving object in the scene or a moving camera to captured images. On the contrary, if the camera is in shutter-priority mode, the shutter speed is fixed while the aperture adjusts its size. Using a wider aperture allows a large amount of light to reach the sensor within a fixed time. However, the DoF is reduced, causing defocus blur in scene regions outside the DoF. However, some computer vision applications require a wide aperture but still want an all-in-focus image, which poses a challenging problem for current devices. An obvious example is cameras on self-driving cars or cameras on cars that map environments. In this case, the camera must use a fixed shutter speed and the only way to get sufficient light is to use the wide aperture, which results in the defocus blur.

An example of defocus blur is shown in Figure 1.6(a). The level of the defocus blur is highly dependent on the scene depth, aperture size and the focal plane of the camera, which means the level of the defocus blur is spatially variant. And a defocus map describes the level of the defocus blur as shown in Figure 1.6(b), which is quantitatively described as the diameter of COC for each pixel. The thin lens model is shown in Figure 1.5 [26].

Based on the thin lens equation $\frac{1}{f} = \frac{1}{f_1} + \frac{1}{S_1}$, where f is the focal length of the lens, S_1 is the focal distance, f_1 is the image distance, the COC diameter c of a scene

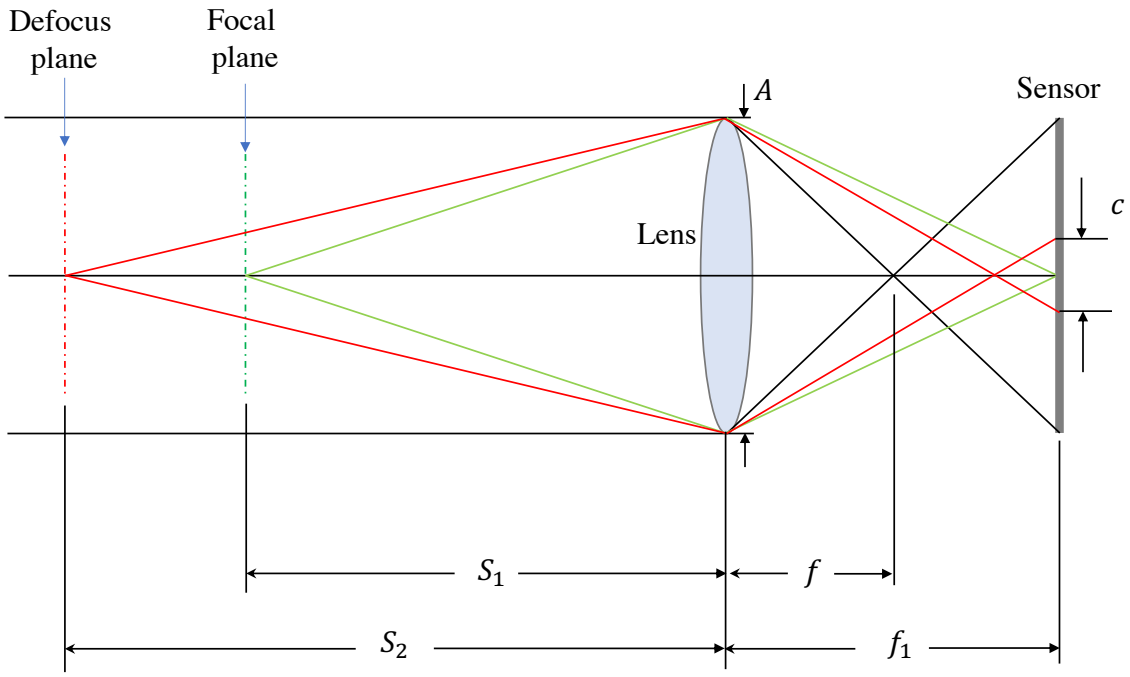


Figure 1.5: Circle of confusion.

point at object distance S_2 can be computed by:

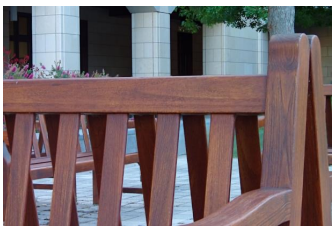
$$c = A\frac{|S_2 - S_1|}{S_2}\frac{f}{S_1 - f} \tag{1.3}$$

where A is the aperture size.

The defocus blurring is an inverse problem - blind deblurring, in which both the blur kernel and the sharp image are unknown. To remove the defocus blur, conventional defocus deblurring methods typically decompose the problem into two steps: defocus map estimation and the non-blind deblurring [27–31]. there still is a CNN-based method that follows first defocus map estimation and then non-blind deblurring [32]. While some end-to-end methods based on convolutional neural network for defocus deblurring are proposed [33–35]. Although the previous works can remove the defocus deblur to some extent, there is still a large gap between the results achieved by these methods and the all-sharp ground truth. It is still a challenging task to achieve defocus blur removal by using a single image. Thus, we pursue more powerful approaches to remove the defocus blur.

(a) Input image

(b) Defocus map



(c) All-sharp ground truth

Figure 1.6: An example for defocus blur.

1.3 Preliminaries

To encourage more understanding, we introduce the preliminary knowledge that is relevant to our work in this section, which includes the convolutional neural network, some architectures related to image quality improvement and the metrics used for evaluation on image quality improvement.

1.3.1 Evaluation Metrics for Image Quality Assessment

Image quality assessment methods are very important for high-fidelity imaging since they show the direction to optimize the designed network by quantitatively measuring the quality of the reconstructed images. Here we will introduce some of them which are used for evaluation in our work.

Mean Square Error (MSE) Mean square error is the commonly-used metric in image quality assessment and it is also well-known as L_2 loss for optimization. It is a reference-based metric, which means it needs ground truth for computation. The value is closer to zero, the performance is better.

Given the reconstructed image $I_{rec} \in \mathbb{R}^{C\times H\times W}$ and the ground truth image $I_{gt} \in \mathbb{R}^{C\times H\times W}$, where the C is the number of channels in the image, $C = 1$ for gray images while $C = 3$ for color images, and H and W are the height and width of the images, respectively, the mean square error between the reconstructed image and the ground truth image is defined as,

$$\mathrm{MSE}(I_{rec}, I_{gt}) = \frac{1}{CWH}\sum_{c=1}^{C}\sum_{x=1}^{W}\sum_{y=1}^{H}(I_{rec}(c,x,y) - I_{gt}(c,x,y))^2, \tag{1.4}$$

Mean Absolute Error (MAE) Mean absolute error is also the commonly-used metric in image quality assessment and it is also well-known as L_1 loss for optimization. It is also a reference-based metric and if the value is closer to zero, the performance is better. The mean absolute error between the reconstructed image and the ground truth image is defined as,

$$\text{MAE}(I_{rec}, I_{gt}) = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{x=1}^{W} \sum_{y=1}^{H} |I_{rec}(c, x, y) - I_{gt}(c, x, y)|, \tag{1.5}$$

Peak Signal-to-Noise Ratio (PSNR) The Peak Signal-to-Noise Ratio is the most commonly-used metric in image quality assessment. The PSNR is an expression for the ratio between the maximum possible signal power and the power of the distorting noise, which affects the quality of its representation. Because many signals (e.g., images) have a very wide dynamic range, The PSNR is usually expressed as the logarithm term of the decibel scale. The PSNR value varies from 30 to 50 dB for image and video quality evaluation due to the 8-bit data representation for images. The PSNR between the reconstructed image and the ground truth image is defined as,

$$\text{PSNR}(I_{rec}, I_{gt}) = 10 \cdot log_{10} \frac{I_{MAX}}{\text{MSE}(I_{rec}, I_{gt}))}, \tag{1.6}$$

where I_{MAX} is the maximum value of an image. For example, for an 8-bit image, the maximum value is 255.

Structure Similarity Index Method (SSIM) The Structural Similarity Index Method is a reference-based perceptual metric. SSIM is computed by the combination of three major components: luminance, contrast and structural [36]. Given two images (or patches) x and y, *luminance* and *contrast* is estimated as the mean and standard deviation of each image,

$$\mu_x = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{1.7}$$

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (x_n - \mu_x)^2} \tag{1.8}$$

where N is the number of pixels in the image x, x_n is the pixel value of n-th pixel location. *Structure* is estimated by their covariance $\sigma_{x,y}$, which is computed by,

$$\sigma_{x,y} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_x)(y_n - \mu_y)} \tag{1.9}$$

Then the SSIM can be defined as,

$$\text{SSIM}(I_{rec}, I_{gt}) = [l(I_{rec}, I_{gt})]^{\alpha} \cdot [c(I_{rec}, I_{gt})]^{\beta} \cdot [s(I_{rec}, I_{gt})]^{\gamma} \tag{1.10}$$

where $l(I_{rec}, I_{gt})$ is the luminance comparison function, $c(I_{rec}, I_{gt})$ is contrast comparison function and $s(I_{rec}, I_{gt})$ is the structure comparison function, respectively, and α, β and γ are positive constants to adjust the effect of luminance, contrast and structure comparison function.

The comparison function are given as

$$l(I_{rec}, I_{gt}) = \frac{2\mu_{rec}\mu_{gt} + C_1}{\mu_{rec}^2\mu_{gt}^2 + C_1} \tag{1.11}$$

$$c(I_{rec}, I_{gt}) = \frac{2\sigma_{rec}\sigma_{gt} + C_2}{\sigma_{rec}^2\sigma_{gt}^2 + C_2} \tag{1.12}$$

$$s(I_{rec}, I_{gt}) = \frac{\sigma_{rec,gt} + C_3}{\sigma_{rec}\sigma_{gt} + C_3} \tag{1.13}$$

where μ_{rec} and μ_{gt} are the local means for the image I_{rec} and I_{gt}, respectively. σ_{rec} and σ_{gt} are the standard deviations for the image I_{rec} and I_{gt}, respectively. And the $\sigma_{rec,gt}$ is the cross-covariance for the image I_{rec} and I_{gt}. When $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$, then the SSIM in Equation 1.10 can be simplified as the following form using Equations 1.11-1.13:

$$\text{SSIM}(I_{rec}, I_{gt}) = \frac{(2\mu_{rec}\mu_{gt} + C_1)(2\sigma_{rec}\sigma_{gt} + C_2)}{(\mu_{rec}^2\mu_{gt}^2 + C_1)(\sigma_{rec}^2\sigma_{gt}^2 + C_2)} \tag{1.14}$$

Learned Perceptual Image Patch Similarity (LPIPS) The Learned Perceptual Image Patch Similarity [37] is also a reference-based perceptual metric. LPIPS

performs a comparison on the features extracted by some deep models, and these models are pre-trained on the large-scale image dataset like ImageNet [7]. As argued in [37], the stronger a feature set is at classification and detection, the stronger it is as a model of perceptual similarity evaluation. Thus, the LPIPS can quantitatively compute the perceptual scores of images based on the features extracted by intermediate layers in the pre-trained deep models.

Given a netowrk, $y^l_{rec} \in \mathbb{R}^{H_l \times W_l \times C_l}$ and $y^l_{gt} \in \mathbb{R}^{H_l \times W_l \times C_l}$ are the normalized features extracted for l-th layer from the reconstructed image I_{rec} and the ground truth I_{gt}. The the LPIPS score between the reconstructed image I_{rec} and the ground truth I_{gt} image is computed as followed,

$$\text{LPIPS}(I_{rec}, I_{gt}) = \sum_{l=1}^{L} \frac{1}{H_l W_l} \sum_{h=1}^{H_l} \sum_{w=1}^{W_l} ||w_l \odot (y^l_{rec} - y^l_{gt})||_2^2 \tag{1.15}$$

where $w_l \in \mathbb{R}^{C_l}$ is the weight vector in l-th layer.

1.3.2 Convolutional Neural Network

Convolutional layers extract image features by performing a weighted summation operation between the convolutional kernel and the input features. Convolutional layers are often stacked in a deep convolutional neural network with the lower convolutional layers extracting simple features such as texture and color and the deeper convolutional layers extracting deeper semantic features. Convolutional layers have the features of weight sharing and local connectivity, which reduce the number of parameters and avoid overfitting. Weight sharing means that the weight parameters of the convolutional kernel are shared for each position in the input features. Compared with the fully connected layer, the weight sharing reduces in the convolutional layer the number of parameters. The local connection means that the convolutional kernel is performed on a local patch in the input features at a time. The advantage of local connectivity is that the number of parameters is reduced. Local connectivity also helps to extract local features of the image.

For simplicity, here we introduce the convolutional operation by setting the number of channel to 1. Given a input feature $x \in \mathbb{R}^{H \times W}$ and a kernel $x \in \mathbb{R}^{K \times K}$, the

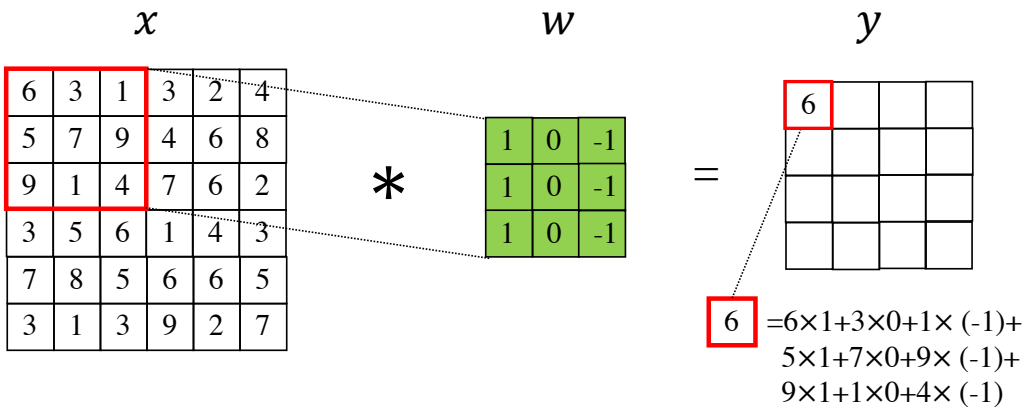


Figure 1.7: An illustration of convolutional operation

convolutional operation is defined as,

$$y(u,v) = \sum_{i=0}^{K}\sum_{j=0}^{K} x(u+i, v+j) \times w(i,j) \tag{1.16}$$

where u and v denote the pixel location of input feature. The convolutional operation is shown in Figure 1.7.

1.3.3 Batch Normalization (BN) Layer

Batch Normalization [38] is initially proposed to reduce the *internal covariate shift* in deep networks. Deep networks consist of many layers and the parameter update of each layer will cause the input distribution of the next layer to change. The input distribution of the deeper layer will change drastically due to a small change in the shallow layer, resulting in a significant shift in deeper layers. So the learning rate, weight initialization, and optimization methods need to be set carefully to train the deep models. Batch Normalization alleviates the *internal covariate shift* problem by adjusting the mean and the standard deviation of the input features for each layer. Specifically, there are two steps in Batch Normalization: the normalization step and the scale and shift step. In the normalization step, the input samples will be normalized to make them have the mean of zero and the standard deviation of 1,

$$\begin{aligned}\mu &= \frac{1}{m}\sum_{i=1}^{m} x_i \\ \sigma^2 &= \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu) \\ \hat{x_i} &= \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}\end{aligned} \tag{1.17}$$

where ϵ is a small constant number to avoid zero division and x_i is the i-th sample in the mini-batch.

Then in the scale and shift step, the normalized features $\hat{x}$ are transformed by,

$$y_i = \hat{x_i} \times \gamma + \beta \tag{1.18}$$

where γ and β are the learnable parameters. Batch Normalization can accelerate the convergence in model training and make the model training process more stable.

1.3.4 Fully Connected (FC) Layer

A fully connected (FC) layer performs a linear combination on each element in a vector by using the learnable parameters, i.e., weight and bias parameters. Given an m dimensional input vector $\mathbf{x} \in \mathbb{R}^m$, the fully connected layer is defined as,

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} + \mathbf{b} \tag{1.19}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the output, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is the weight parameter, and $\mathbf{b} \in \mathbb{R}^n$ is the bias parameter. An illustration of FC layer is shown in Figure 1.8

1.3.5 Pooling Layer

The pooling layer is used to reduce the spatial dimension of the input. Then the non-representative features will be filtered out. Thus the more valuable features can be extracted. The commonly-used pooling layers are *Max pooling* and *Average pooling* as shown in Figure 1.9.

Given a feature map $F \in \mathbb{R}^{H \times W \times C}$, a max pooling layer with stride s and kernel

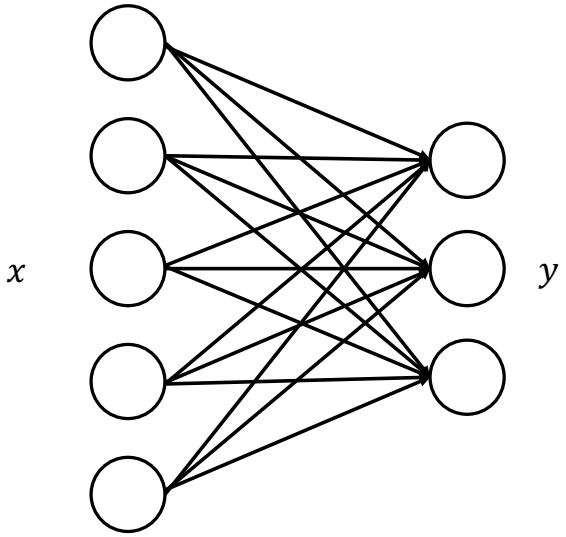


Figure 1.8: Fully connected layer

size K is applied to each c-th channel of F as follows,

$$\begin{aligned} O(i,j,c) =& \max(F(s\times i+1, s\times j+1, c)), \ldots, \\ & F(s\times i+m, s\times j+n, c)), \ldots, F(s\times i+K, s\times j+K, c)) \end{aligned} \tag{1.20}$$

For an average pooling layer, the output is computed as follows,

$$O(i,j,c) = \frac{1}{K\times K}\sum_{m=1}^{K}\sum_{n=1}^{K} F(s\times i+m, s\times j+n, c)) \tag{1.21}$$

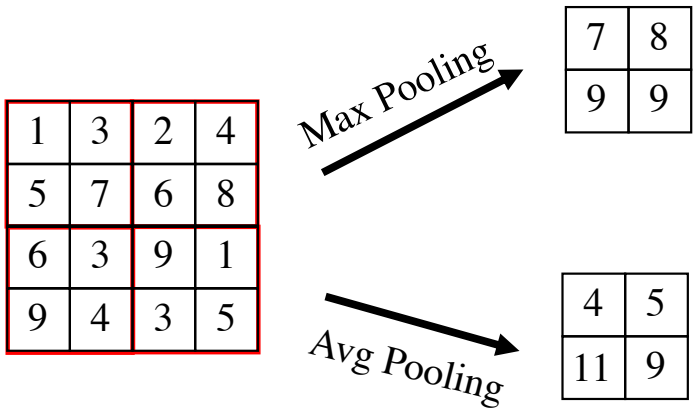


Figure 1.9: Two typical pooling layers, i.e. max pooling and average pooling with stride 2 and kernel size 2.

1.3.6 Activation Function

Activation functions play a very important role in providing the non-linear modeling capacity of the deep neural network since the convolutional operation can be considered as matrix multiplication which is only a linear combination of the input features or images, while the activation functions provide the non-linear mapping for the features. We introduce some commonly-used activation functions as follows.

Sigmoid Function or Logistic Function Sigmoid function is the commonly-used activation function that has "S"-shaped curve. Sigmoid function maps arbitrary input of real value into the probability-like output, $[0, 1]$, as shown in Figure 1.10. The formulation of sigmoid function is defined as

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \tag{1.22}$$

One main disadvantage of sigmoind function is that it is computationally expensive since it involves exponential in nature. And since the output of sigmoind function is in $[0, 1]$, it is a non-zero-centered function, which leads to slow convergence.

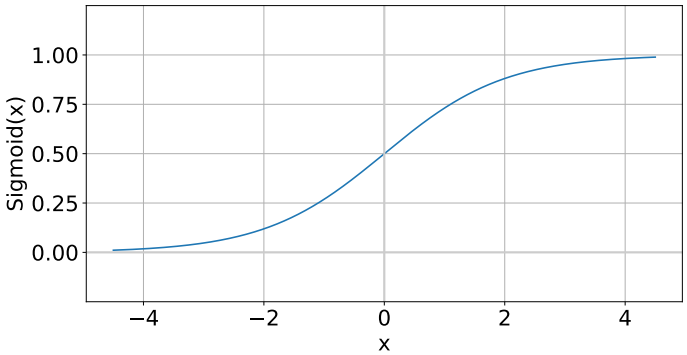


Figure 1.10: The plot of sigmoid function.

Hyperbolic Tangent Function While hyperbolic tangent function (tanh) is a zero-centered function since its output range is in $[-1, 1]$. This plot between input and output is shown in Figure 1.11. The function is defined as

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}. \tag{1.23}$$

Since tanh also involves exponential in nature, the computational cost of tanh is also high.

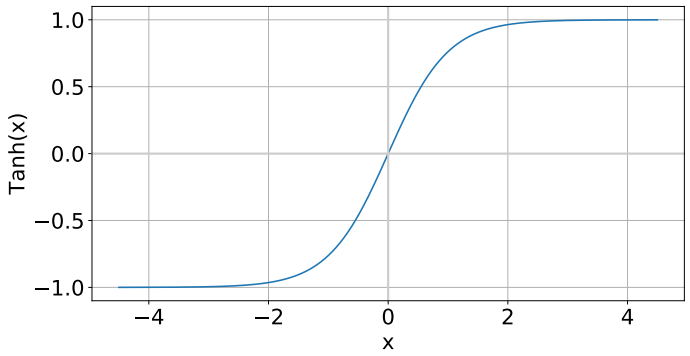


Figure 1.11: The plot of hyperbolic tangent (tanh) function.

Rectified Linear Unit (ReLU) [39] The above two functions have been successfully used in the neural network. However, when the above two functions are used in the hidden layers, *gradient vanishing* would commonly occur. This is because when the absolute value of the input is extremely large, the above two functions will saturate.

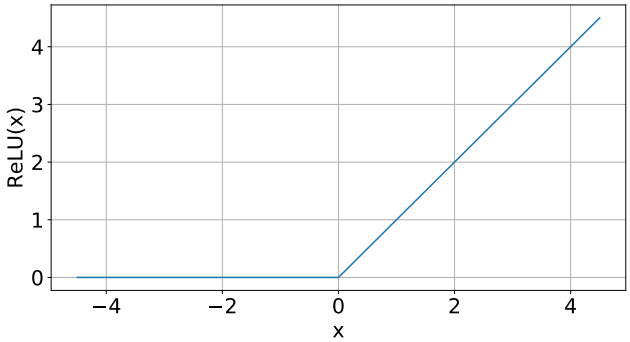


Figure 1.12: The plot of rectified linear unit (ReLU) function.

The ReLU function can avoid the *gradient vanishing* problem and it is defined as

$$\text{ReLU}(x) = max(0, x). \tag{1.24}$$

Although the ReLU function looks like a linear function, it is a non-linear function since the output is zero when the input is smaller than zero, while the output is

the same as the input when the input is higher than zero. Due to the simplicity of ReLU, the computational cost is low. And the ReLU function causes the output of some neurons to be zero, which imposes the sparsity on the network, alleviating the overfitting problem.

Softmax The softmax function takes a vector with N real numbers as input and outputs a probability distribution. The softmax function is defined as,

$$\text{Softmax}(x)_i = \frac{e^{x_i}}{\sum_{n=1}^{N} e^{x_n}}, \text{ for } i = 1, \ldots, N. \tag{1.25}$$

Thus, the each element in the output of softmax is in $[0, 1]$ and the over all elements will add up to 1.

1.3.7 Deep Residual Learning

Deep residual learning is introduced by [6]. An obvious problem on training deep models is the gradients vanishing and/or exploding [6], which hammer the convergence of the training. Proper parameters initialization [40, 41] and normalization layer [38] in the network can largely address this problem. Another problem for deep models is the *degradation*: the performance gets saturated with the deeper models. Without a proper setting, the performance of the deep models is limited. By introducing residual learning, the deep models can be trained and achieve much better performance. Residual learning is defined as,

$$y = \mathcal{F}(x) + x, \tag{1.26}$$

where x is the input features and $\mathcal{F}(.)$ is the function implemented by some convolutional layers or blocks. Figure 1.13 shows an illustration of residual learning, in which $\mathcal{F}(.)$ is implemented by two convolutional layers.

Loss Function and Gradient Descent Optimization

The loss functions commonly-used in image enhancement tasks are L_1 loss, L_2 loss, SSIM loss, perceptual loss [42] and adversarial loss [12]. The model trained on L_2

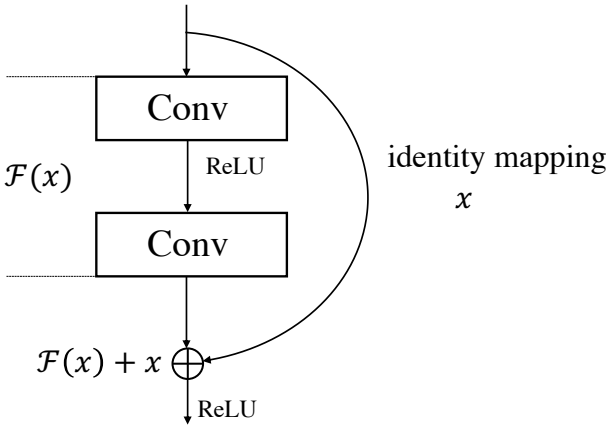


Figure 1.13: An illustration of residual learning.

loss achieves the blurry images since the L_2 loss correlates poorly with image quality as perceived by a human observer [43] and works on the Gaussian noise distribution assumption, which is not the truth for many tasks. While the model trained on L_1 loss achieves the sharper images in practice. The perceptual loss and adversarial loss, on the other hand, can provide more details for the reconstructed image.

Training the deep neural network needs the paired data, i.e., the low-quality images and the corresponding ground truth image in the image-to-image translation task. The low-quality images are fed into the network to generate the reconstructed images and then the loss is computed by the loss functions between the reconstructed image and the corresponding ground truth image.

The optimization methods adjust the parameters iteratively to reduce the loss. These methods are the *gradient descent optimization* methods. There are some commonly-used optimization methods in deep learning, for example, SGD [44] and Adam [45]. In general, gradient descent methods update the parameters in each step using the mini-batch with n data samples by,

$$w^{i+1} = w^i - \eta \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla_w L_i \tag{1.27}$$

where w is the learnable parameters, η is the step size or the learning rate, L_i is the loss computed on the i-th sample, $\nabla_w L_i = \frac{dL_i}{dw}$ is the gradient with respect to the parameter w on loss L_i.

1.4 Outline of the Dissertation

In this dissertation, we consider the quality improvement of the imaging from limited devices. Specifically, we try to improve the quality in the high dynamic range imaging and the defocus image deblurring. The outline of the content is described in what follows.

Chapter 2 We consider the problem of generating an HDR image of a scene from its LDR images. Recent studies employ deep learning and solve the problem in an end-to-end fashion, leading to significant performance improvements. However, it is still hard to generate a good quality image from LDR images of a dynamic scene captured by a hand-held camera, e.g., occlusion due to the large motion of foreground objects, causing ghosting artifacts. The key to success relies on how well we can fuse the input images in their feature space, where we wish to remove the factors leading to low-quality image generation while performing the fundamental computations for HDR image generation, e.g., selecting the best-exposed image/region. We propose a novel method that can better fuse the features based on two ideas. One is multi-step feature fusion; our network gradually fuses the features in a stack of blocks having the same structure. The other is the design of the component block that effectively performs two operations essential to the problem, i.e., comparing and selecting appropriate images/regions. Experimental results show that the proposed method outperforms the previous state-of-the-art methods on the standard benchmark tests.

Chapter 3 In addition, we further consider the problem of the reconstruction of ghosting-free HDR images of dynamic scenes in the alignment-before-merging approach from a set of multi-exposure images, especially with large object motion and occlusions. To address this problem, we propose a deep network that tries to learn multi-scale feature flow guided by the regularized loss. It first extracts multi-scale features and then aligns features from non-reference images. After alignment, we use residual channel attention blocks to merge the features from different images. Extensive qualitative and quantitative comparisons show that our approach achieves state-of-the-art performance and produces excellent results where color artifacts and geometric distortions are significantly reduced.

Chapter 4 Finally, we consider the problem in defocus image deblurring. Previous

classical methods follow two-step approaches, i.e., first defocus map estimation and then the non-blind deblurring. In the era of deep learning, some researchers try to address these two problems by CNN. However, the simple concatenation of defocus map, which represents the blur level, leads to suboptimal performance. Considering the spatial variant property of the defocus blur and the blur level indicated in the defocus map, we employ the defocus map as conditional guidance to adjust the features from the input blurring images instead of simple concatenation. Then we propose a simple but effective network with spatial modulation based on the defocus map. To achieve this, we design a network consisting of three sub-networks, including the defocus map estimation network, a condition network that encodes the defocus map into condition features and the defocus deblurring network that performs spatially dynamic modulation based on the condition features. And the spatially dynamic modulation is based on an affine transform function to adjust the features from the input blurry images. Experimental results show that our method can achieve better quantitative and qualitative evaluation performance than the existing state-of-the-art methods on the commonly used public test datasets.

Chapter 2

Progressive and Selective Fusion Network for High Dynamic Range Imaging

2.1 Introduction

As real-world scenes usually have a range of luminosity beyond the dynamic range of imaging devices, standard digital cameras can only produce low dynamic range (LDR) images containing under-exposure and over-exposure regions where the detailed information is missing. There are demands for high dynamic range (HDR) imaging from various fields such as movie [46] and computer rendering [47], etc. There are special cameras that can capture HDR images, which tend to be expensive. Thus, there are methods for generating an HDR image from a series of LDR images captured by a standard camera with different exposure settings. While they can produce high-quality HDR images for static scenes, these methods tend to yield images with many ghosting artifacts for dynamic scenes or even for static scenes when the input images are captured by a hand-held camera. Numerous efforts have been made so far to remove the ghosting artifacts in the HDR image reconstruction. There are several methods that attempt to detect motion regions in the input LDR images and then remove these regions [48–50]. However, they tend to work well only when the motion in the input images is relatively small. When there is large motion, a large

number of image pixels need to be removed, which results in incorrect reconstruction due to missing information about these pixels. There are also methods that align the input LDR images using the optical flows to a reference image from the others before merging them [51–53]. While these methods can handle larger image motion, their performance highly depends on the accuracy of the estimated optical flows. When the motion regions are either over or under-exposed, there tend to emerge noticeable artifacts in the resulting HDR images.

More recent studies employ convolutional neural networks (CNNs) and formulate the problem in an end-to-end fashion. They train CNNs to learn the direct mapping from multiple LDR images to an HDR image using appropriate training data [5,54,55]. Although they achieve better performance than the above methods, they still suffer from the ghosting artifacts when there is large motion in the input LDR images.

We can think of the current CNN-based methods employing the same approach to the problem, i.e., fusing the input images in their feature space and then reconstructing an HDR image from the fused feature, typically using an encoder-decoder network. It relies on the feature fusion to solve the two fundamental problems, i.e., selecting well-exposed images/regions from the input images and correcting/eliminating the misalignment of the images plus possible occlusions due to object/image motion. Previous studies perform this feature fusion in a single step using a relatively simple method such as summation and concatenation. We think this leads to suboptimal feature fusion, causing ghosting artifacts.

In this chapter, we propose a novel network named *progressive and selective feature network* (PSFNet) to resolve the above issue. PSFNet employs i) a multi-step approach that progressively fuses features and ii) a more suitable mechanism for feature fusion.

For (i), we split the difficult problem of feature fusion into multiple steps, by which we intend to make it easier for the network to learn to solve it. It is analogous to nonlinear optimization algorithms that update parameters iteratively. PSFNet is designed to fuse the image features in a progressive fashion using a stack of blocks having the same structure named the *progressive and selective feature block* (PSFB). A single PSFB updates the image features by a small amount, and a series of PSFBs updates them gradually, completing their fusion.

For (ii), we design the PSFB that effectively performs the two operations playing the central roles in the feature fusion, i.e., *comparison* and *selection*. The former compares the input LDR images to identify their differences. The latter selects the images/regions based on the comparison results. These two operations are the key to successful HDR image computation since it is fundamentally correctly selecting images/regions that are well-exposed and properly eliminating inter-image misalignment and possible occlusions.

The experiments demonstrate that the above approach can successfully produce ghosting-free HDR images. Our method can achieve better performance in terms of quantitative and qualitative evaluation than the popular algorithms on the commonly used public test datasets.

2.2 Pioneering Works

Motion Detection and Removal Some methods classify all the pixels of the input LDR images as static or moving depending on whether they correspond to static background or moving foreground. They then merge the pixels found to be static while removing those found to be moving. In [48], weights are computed iteratively and then applied to pixels to determine their contribution to the final image. Heo et al. [49] propose a method that assigns a weight to each pixel by computing a Gaussian-weighted distance to a reference pixel. Jacobs et al. [56] propose a method that detects moving pixels by calculating the variance of different LDR images. Zhang et al. [57] propose to detect image motion by analyzing image gradients. Rank minimization [58] and sparse representation [50] have been employed to detect outliers, including moving pixels, and reconstruct the final HDR image. However, even when all pixels are classified correctly, removing the moving pixels makes it impossible to utilize all the information contained in the input images; the generated HDR images will inevitably lose some details.

Alignment based Methods These methods first align the input LDR images and then merge them to reconstruct an HDR image. The images are aligned either at the pixel level or the patch level. In [51], the optical flow field is estimated and used for the alignment by warping the input images with them. In [59], a method for merging

images is proposed to eliminate the artifacts of the alignment using optical flow. Sen et al. [2] formulate the HDR reconstruction as an energy-minimization task that jointly solves patch-based alignment and HDR image reconstruction. Hu et al. [52] propose a method for image alignment using brightness and gradient information. Hafner et al. [53] propose an energy-minimization method that simultaneously computes the aligned HDR composite and accurate displacement maps. These methods tend to fail to deal with large motion and excessively dark or bright regions since the alignment process is vulnerable to them, generating artifacts in the aligned images. These methods employ a simple method for merging aligned LDR images, which cannot eliminate those artifacts.

Deep Learning based Methods As with other similar tasks, deep learning has been applied to HDR image generation. Eilertsen et al. [60] propose a deep network having the encoder-decoder structure for HDR image generation from a single image. It is proposed in [61] to synthesize multiple LDR images with different exposures from a single LDR image with a deep learning based method and use them to generate an HDR image. Such single-image-based methods tend to fail to reconstruct the textures on the saturated regions accurately. Kalantari et al. [1] propose using a convolutional neural network (CNN) for the task, which takes the LDR images for the input aligned in advance using optical flow. Wu et al. [5] propose to use a CNN with the U-net structure to reconstruct a ghosting-free HDR image without explicit alignment of the input images. Yan et al. [55] propose a non-local structure into a U-net to improve the accuracy of HDR image generation. Yan et al. [54] propose attention modules for improving the merging of input images with a reference image. Pu et al. [62] use deformable convolution [63] across multiscale features to perform pyramidal alignment of input images and also use attention mechanism to reconstruct an HDR image from the aligned feature accurately.

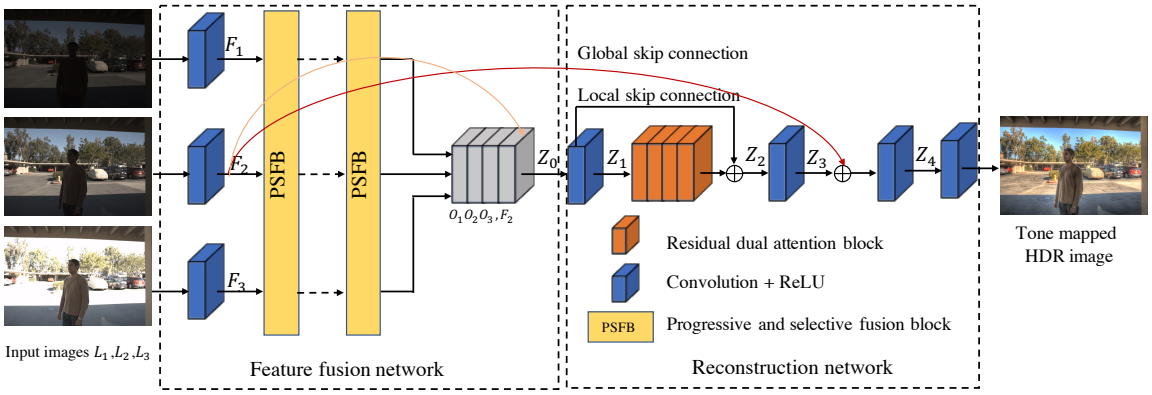


Figure 2.1: Archtecture of proposed network.

2.3 Proposed Method

2.3.1 Outline of the PSFNet

Given a set of LDR images, $L_1, L_2, ..., L_k$, of a dynamic scene with different exposures, the goal of HDR imaging is to reconstruct an HDR image H aligned to a selected reference image L_r. Following the settings in [1], we consider the case of using three LDR images, L_1, L_2, and L_3, sorted in the order of exposures. We select L_2 as the reference image. Following [1], before feeding them into the network, we map the LDR images into an HDR domain using gamma correction. To be speicfic, we map L_i to H_i as

$$H_i = L_i^{\gamma}/t_i, \quad i = 1, 2, 3, \tag{2.1}$$

where $\gamma = 2.2$ [64] and t_i is the exposure time of L_i. We then concatenate L_i and H_i in the channel dimension to get a six channel tensor $X_i = [L_i, H_i]$ for each of $i = 1, 2, 3$ and input X_1, X_2, and X_3 to our network. We expect that L_i's help identify image noises and/or saturated regions, whereas H_i's help identify the differences from the reference image.

Our network consists of two sub-networks, *the feature extraction network* and *the reconstruction network*, as shown in Figure 2.1. While the overall construction is similar to the encoder-decoder design employed in previous studies [54, 62], we design the sub-networks with clearer intentions. Specifically, the feature fusion network fuses the input LDR images in their feature space, aiming to eliminate their misalignment and

possible occlusions due to moving objects. The reconstruction network reconstructs an HDR image from the fused feature. We will explain their details in what follows.

2.3.2 Feature Fusion Network

The input LDR images are geometrically unaligned and could contain occlusions, which need to be eliminated. Previous studies leave it to the fusion of the input image features in the encoder part of a network, which is also intended to generate an HDR image from multiple input images. Most of them perform the feature fusion in a single step using a simple operation such as summation and concatenation. We think this leads to suboptimal results causing ghosting artifacts etc. For better feature fusion, we employ i) a multi-step approach that progressively fuses features and ii) a more suitable mechanism for the feature fusion. We explain the two below.

Progressive Feature Fusion

Initially extracting features from the input LDR images, the feature fusion network fuses their features in a progressive fashion using a stack of blocks having the same structure, as shown in Figure 2.1. We name the block *the progressive and selective fusion block* (PSFB). In our experiments, we use a stack of six PSFBs.

Our intention behind this design is to split the difficult task of feature fusion into multiple steps, by which we attempt to make it easier for the network to learn to perform the task. It is analogous to nonlinear optimization algorithms that update parameters iteratively. Previous studies of other tasks employed this idea of designing an architecture to gradually update estimates to get better results, e.g., RAFT (recurrent all-pairs field transforms) for optical flow estimation [65]. Our approach shares the same motivation.

Progressive and Selective Feature Block

Roughly speaking, the computation of an HDR image requires to conduct the following two: the selection of images/regions that are well-exposed (i.e., neither under nor over-exposed) and the elimination of inter-image misalignment and possible

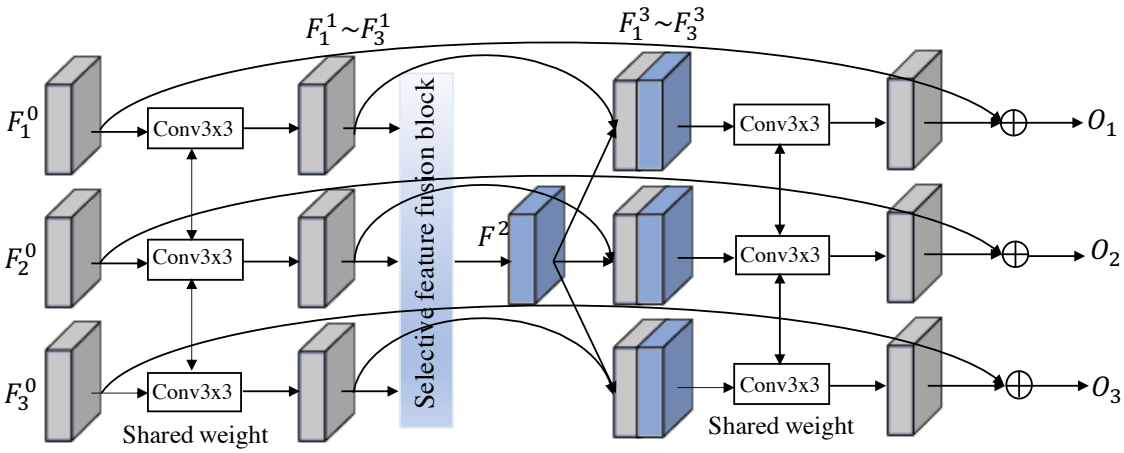


Figure 2.2: Progressive and selective fusion block (PSFB).

occlusions. For the latter, it will be necessary first to identify differences among the images.

These computations will essentially reduce to two fundamental operations, *comparison* and *selection*, i.e., comparing the images to identify their differences and selecting them based on the comparison results. We design the PSFB to perform these two operations effectively; see Figure 2.2.

A PSFB performs the inter-image comparison in its second half. Specifically, we concatenate the fused feature computed in its first half with the individual image features and then apply convolution to each of the concatenated features, as shown in Figure 2.2. This computation will perform the above comparison since the fused feature should contain all the image information; the convolution will learn to compare the input image features and identify their differences.

The PSFB performs the second operation of *selection* in a component named the *selective feature fusion block* (SFFB) in its first half. To design the SFFB, we borrow the feature fusion mechanism of the selective kernel convolution [66], which was developed to adaptively choose the size of convolution kernels (e.g., 3×3 or 5×5) in a convolution layer. Figure 2.3 shows the design of SFFB. It receives inputs from three input features from different images. We first combine these features using an element-wise summation as: $F = F_1^1 + F_2^1 + F_3^1$. Then global average pooling is applied across the spatial dimension of $F \in \mathbb{R}^{H \times W \times C}$ to generate channel-wise statistics as $s \in \mathbb{R}^{1 \times 1 \times C}$. A compact feature vector $z \in \mathbb{R}^{1 \times 1 \times d}$ is achieved by a

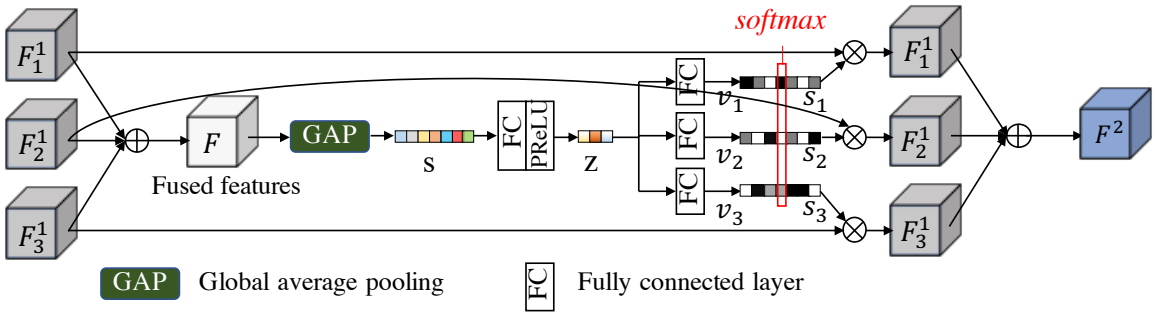


Figure 2.3: Selective feature fusion block (SFFB).

simple fully connection (fc) layer with dimensionality reduction. d is set to $\frac{C}{8}$ in our experiments. Finally, the feature vector z passes through three parallel fc layers with the dimensionality expansion to generate three feature de scriptors v_1, v_2 and v_3, each with dimensions $1 \times 1 \times C$. Softmax function is applied to v_1, v_2 and v_3, generating attention weights s_1, s_2 and s_3 that we use to adaptively recalibrate the feature maps from different images, respectively. The final feature map F^2 is obtained through the attention weights on feature maps from different images: $F^2 = s_1 \cdot F_1^1 + s_2 \cdot F_2^1 + s_3 \cdot F_3^1$.

Previous studies fuse multiple features with simple methods such as summation and concatenation, which we think limits the expressive power of the network. Following [66], SFFB performs dynamic fusion via two operations, i.e., *fuse* and *select*. The fusion operator aggregates the input features by their summation followed by global average pooling and two fully-connected layers, yielding attention weights on the channels of the individual input features. The selection operator applies these attention weights to the input features. The attended features are added to form a fused feature, which is the final output of a SFFB.

It should be noted that while the order of comparison and selection is exchanged within a PSFB, the fused feature is necessary for the comparison of the image features as above, and their intra-block order does not matter since we stack multiple PSFBs as mentioned above.

Details of Computation in PSFB

Taking the first block as an example, we explain the detailed design of a PSFB here. Firstly, setting the input feature $F_i^0 = F_i$ (as this is the first block), it updates

F_i^0 into F_i^1 using a convolution layer with kernel size of 3×3 as

$$F_i^1 = Conv1(F_i^0). \tag{2.2}$$

We then fuse features $\{F_1^1, F_2^1, F_3^1\}$ using the above SFFB, yielding a fused feature F^2 as

$$F^2 = SFFB([F_1^1, F_2^1, F_3^1]). \tag{2.3}$$

We concatenate the fused feature F^2 with F_i^1 as $F_i^3 = [F^2, F_i^1]$ $(i = 1, 2, 3)$. Thus, F_i^3 contains the individual image feature and the fused feature. We use a convolution layer with the kernel size of 3×3 to convert it to an feature O_i as

$$O_i = Conv2(F_i^3) + F_i^0 \tag{2.4}$$

We use "$+F_i^0$" to represent a residual connection [6]; the output of $Conv2()$ has the same size as F_i^0.

Summary of the Feature Fusion Network

As shown in Figure 2.1, the input to the feature fusion network are the three 6-channel input images X_1, X_2, and X_3 corresponding to the input LDR images. The network first extracts a feature map with $N = 64$ channels from each using a shared convolutional layer, yielding F_1, F_2, and F_3. These are inputted to the stack of six PSFBs. As explained above, a single PSFB updates the image features by a small amount, and thus the features are progressively fused in the PSFB stack. The input and output of all the PSFBs have the same size and channels. The features outputted from the last PSFB are concatenated as $Z_0 = [O_1, O_2, O_3, F_2]$ and then inputted to the reconstruction network.

2.3.3 Reconstruction Network

In [54], three dilated residual dense blocks are used to decode the encoded image feature, reconstructing an HDR image. Although their method achieves good performance, it tends to consume a large amount memory, especially when the input image size is large. To cope with this, we employ a residual block with dual attention

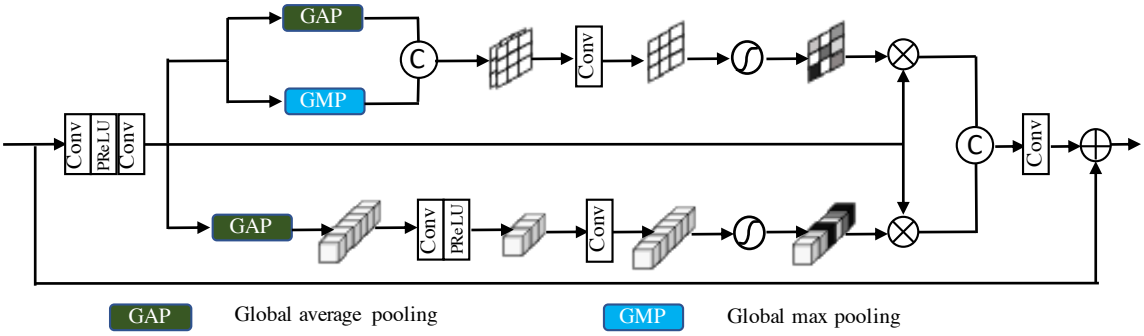


Figure 2.4: Dual attention block

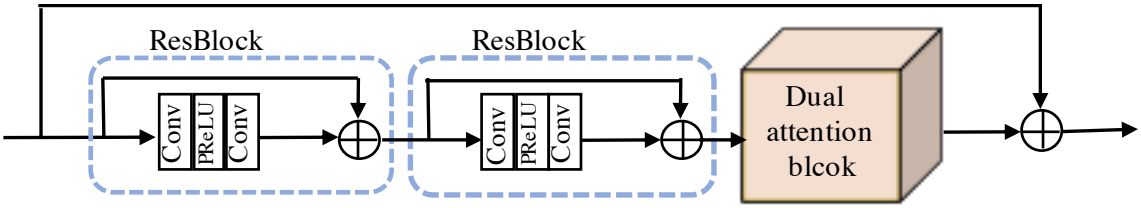


Figure 2.5: Residual dual attention block

mechanism [67] consisting of channel attention [68] and spatial attention [69]. The residual dual attention mechanism is known to work well for superresolution [69] and denoising [67]. We expect it to work for our case because of their similarity.

As shown in Figure 2.1, the reconstruction network takes the concatenated features $Z_0 = [O_1, O_2, O_3, F_2]$. It consists of a series of a convolution layer, four residual dual attention blocks, and three additional convolution layers with a local and a global skip connections. The structure of a dual attention block is shown in Figure 2.4 and the structure of a residual dual attention block is shown in Figure 2.5. Applying a sigmoid function to the output of the last convolution layer, the reconstruction network outputs an HDR image.

2.3.4 Loss Function

Following [1], we consider the optimization in the domain of tone-mapped HDR images. It produces better results with fewer artifacts in the dark regions than the optimization in the original domain of HDR images. To be specific, we employ μ-law

Table 2.1: Comparison of the methods on the test set of [1]. The primary metrics are PSNR-μ, SSIM-μ, and HDR-VDP-2; see Sec. 3.4.1 for more detailss.

Methods	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L	HDR-VDP-2
TMO [61]	8.3120	8.8459	0.5029	0.0924	44.3345
HDRCNN [60]	13.7054	13.8956	0.5924	0.3456	47.5690
Sen [2]	40.9689	38.3425	0.9859	0.9764	60.3463
Kalantari [1]	42.7177	41.2200	0.9889	0.982	61.3139
Wu [5]	41.9977	**41.6593**	0.9878	0.9860	61.7981
AHDR [54]	43.7013	41.1782	0.9905	0.9857	62.0521
PAN [62]	43.8487	41.6452	0.9906	**0.9870**	62.5495
PSFNet	**44.0613**	41.5736	**0.9907**	0.9867	**63.1550**

for tone mapping loss, which is formulated as,

$$T(H) = \frac{log(1+\mu H)}{log(1+\mu)}, \tag{2.5}$$

where μ is set to $5,000$ throughout our experiments. It is also reported in [54] that minimizing the L_1 norm between the predicted HDR image $\hat{H}$ and its ground truth H in the tone-mapped domain works better than others. Following their study, we employ the following loss

$$L = \|T(\hat{H}) - T(H)\|_1. \tag{2.6}$$

2.4 Experiments

2.4.1 Experimental Settings

Training data

We train our network on the dataset of Kalantari and Ramamoorthi [1], which consists of indoor and outdoor scenes. It includes 74 samples for training and 15 samples for testing. We use the former for training the PSFNet. Each sample includes

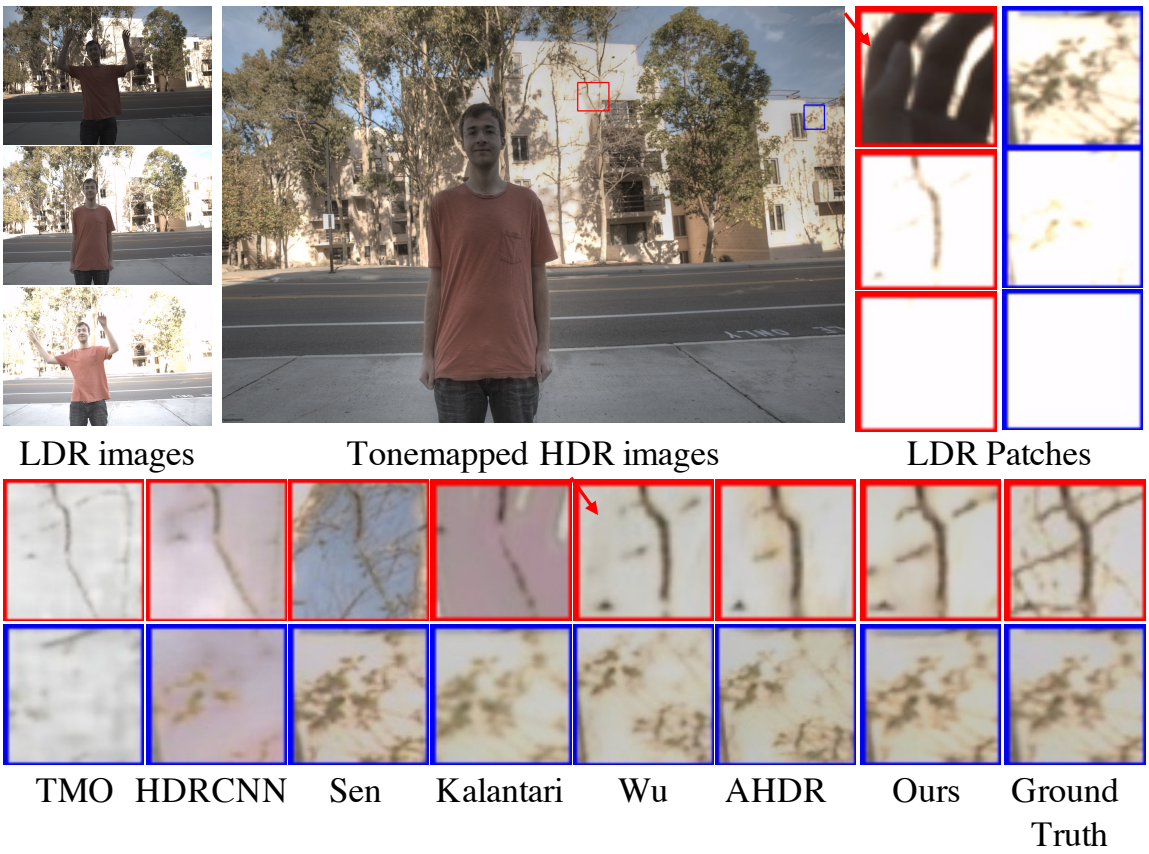


Figure 2.6: Results for "Building" from the test set of [1]. Upper row from left to right: the three input LDR images, the HDR image produced by the proposed method, and (zoomed-in) LDR image patches with two identical positions/sizes (in red and blue). Lower row: the same patches of the HDR images produced by different methods.

ground truth HDR images and three LDR images with exposure biases of $\{-2, 0, +2\}$ or $\{-3, 0, +3\}$. Following the standard procedure of recent studies [1,54,62], we resize all the images to $1,000 \times 1,500$ pixels.

Testing data

Following recent studies, we choose the datasets for testing. We evaluate methods on the 15 scenes of the dataset of [1]; we conduct quantitative evaluation using the provided ground truths. We also use the datasets of Sen et al. [2] and Tursun et al. [3], which do not contain ground truths; we use them for qualitative evaluation.

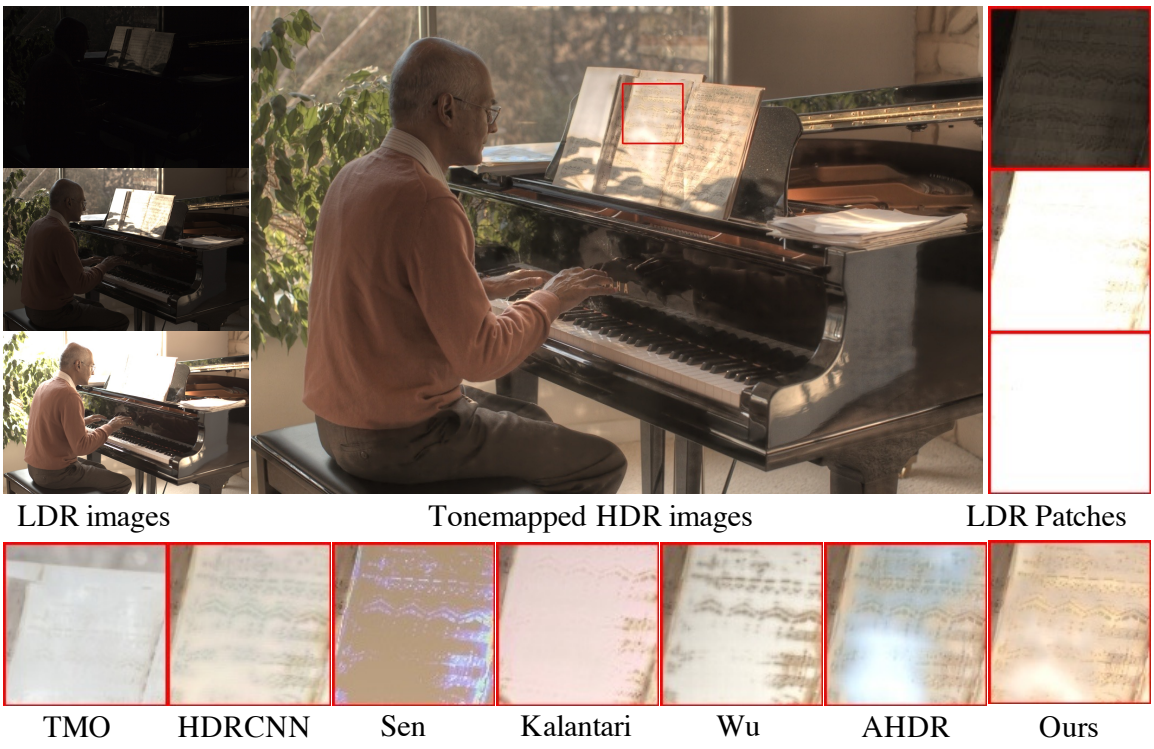


Figure 2.7: Results for "PianoMan" from the dataset of [2]. See Figure 2.6 for the explanation of the panels.

Evaluation metrics

It is argued in [1] that HDR image generation methods should be evaluated in the tone-mapped domain, as we usually use generated HDR images after tone-mapping. Following this argument, we use PSNR-μ and SSIM-μ for primary metrics, which are PSNR and SSIM values in the tone-mapped domain. We show PSNR and SSIM in the linear domain, denoted by PSNR-L and SSIM-L, for completeness. We also show HDR-VDP-2 [70], which is designed to evaluate the quality of HDR images.

It should be noted that there is a limitation in the evaluation based on PSNR-μ etc. The recent studies, including ours, aim at adequately dealing with ghosting artifacts. However, the artifacts usually appear in a small area of an image, and they often have only small impacts on these metrics. HDR-VDP-2 may better evaluate the image quality in that case. To supplement the quantitative evaluation, we also show the results of qualitative comparisons.

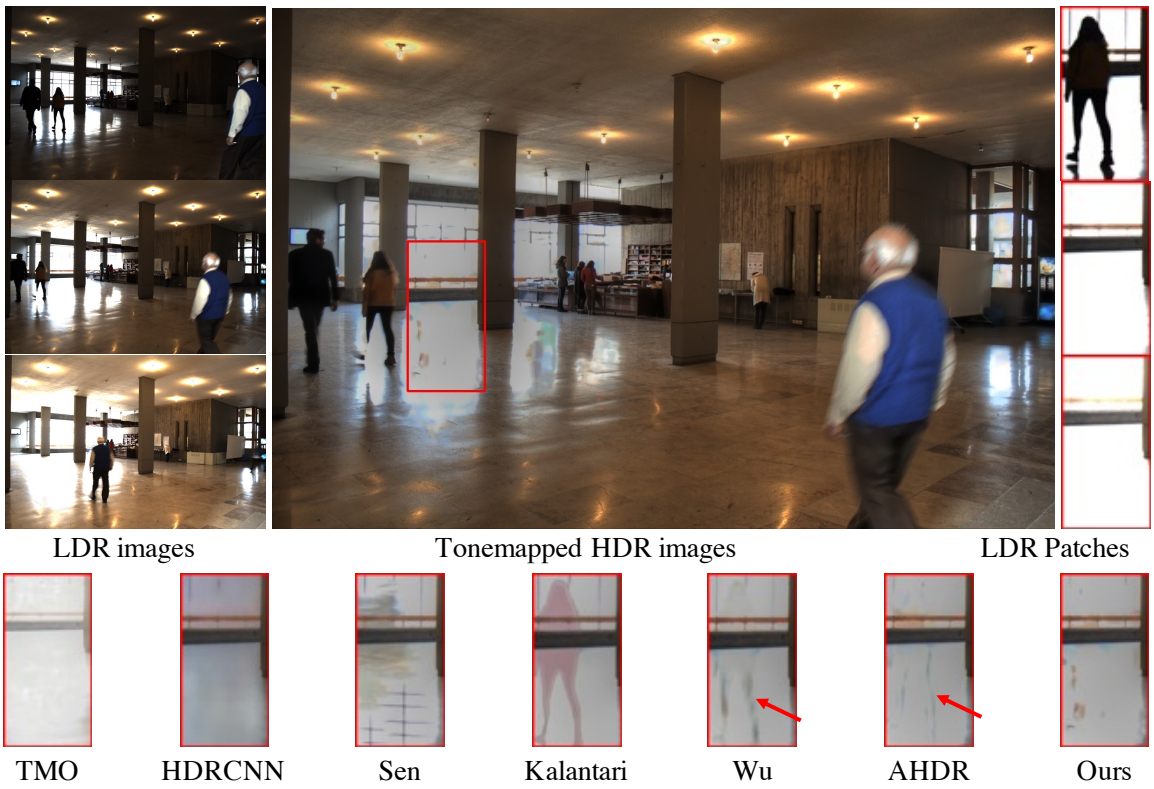


Figure 2.8: Results for an image from the dataset of [3]. See Figure 2.6 for the explanation of the panels.

2.4.2 Implementation Details

The training data are first cropped into patches of 256×256 pixel size with stride of 128 pixels. We employ rotation and flipping for data augmentation to avoid overfitting. We use the Adam optimizer [45] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, initial learning rate $= 10^{-4}$ and set the batch size to 8. We perform training for 210 epochs and employ the cosine annealing strategy [71] to steadily decrease the learning rate from initial value to 1×10^{-6}. We conducted all the experiments using PyTorch [72] on NVIDIA GeForce RTX 2080 GPUs.

2.4.3 Comparison with the State-of-the-art Methods

We evaluate the proposed method and compare it with previous methods including the state-of-the-art. We use Kalantari's testset [1] for quantitative evaluation and the above datasets without ground truths [2, 3] for qualitative evaluation. The compared

Figure 2.9: Results from the Testing data (08) of [1]. See Figure 2.6 for the explanation of the panels

methods are as follows: two single image HDR imaging methods, TMO [61] and HDRCNN [60], and five multi-image HDR imaging methods, the patch-based method [2], the flow-based method with CNN merger [1], the U-net structure without optic flow [5], the attention-guide method (AHDR) [54], and pyramidal alignment network (PAN) [62]. For all methods, we used the authors' code for testing comparison, except [62] since their code is not available as of the time of writing this paper.

Evaluation on Kalantari et al.'s Dataset

Table 3.1 shows the quantitative evaluation on the test set of [1], i.e., averaged values over 15 test scenes. It is seen that the proposed method achieves better performance than others in the primary metrics, PSNR-μ, SSIM-μ, and HDR-VDP-2. Figure 2.6, Figure 2.9, Figure 2.10 and Figure 2.11 show examples of qualitative comparisons. The input LDR images contain saturated background and foreground motions. It is observed from the results of the single-image methods, TMO [61] and

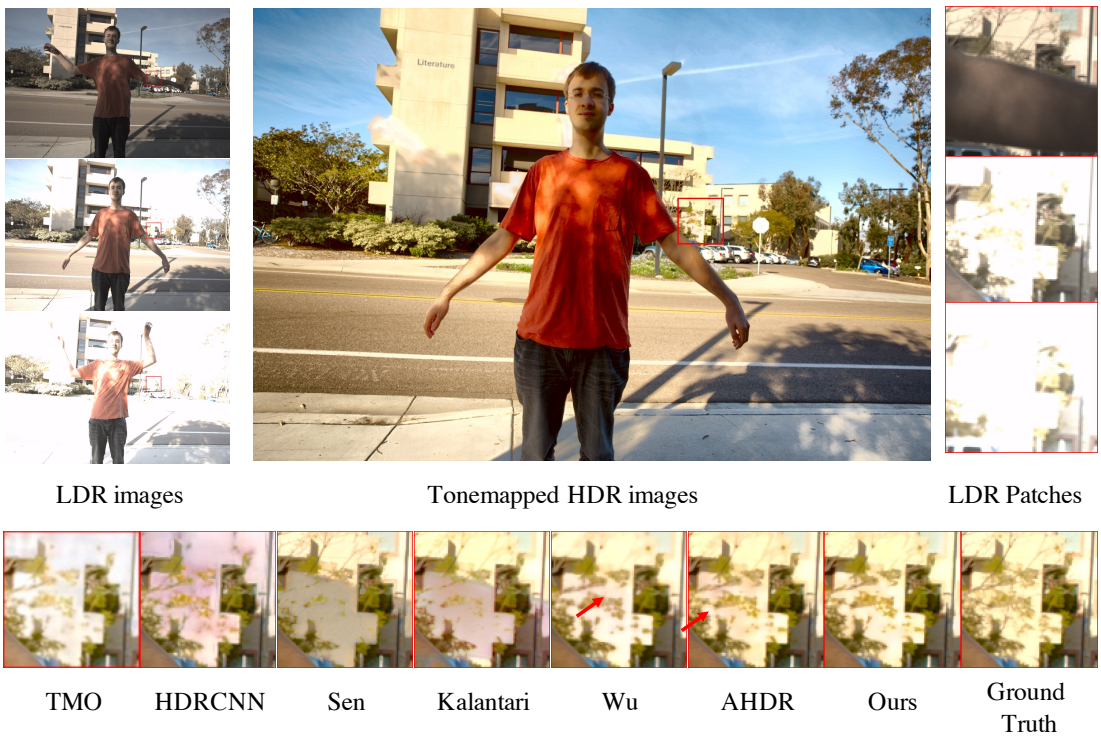


Figure 2.10: Results from the Testing data (09) of [1]. See Figure 2.6 for the explanation of the panels

HDRCNN [60], which uses the reference image alone, that while they can avoid the ghosting artifacts, they cannot recover detailed textures; they also suffer from color distortion. The patch based method of Sen et al. [2] fails to find correct patches, generating some artifacts. The method of Kalantari et al. [1] cannot completely eliminate the effects of the occlusion. The method of Wu et al. [5] and AHDR [54] produce better results but fail to recover the fine details of the texture. Our method produces the best results; it produces less color distortion and recovers the textures more accurately.

We also show several samples from from the test set of [1].

As shown in Figure 2.12 and 2.13, there are some samples that our method can produce almost the same output with the groudtruth.

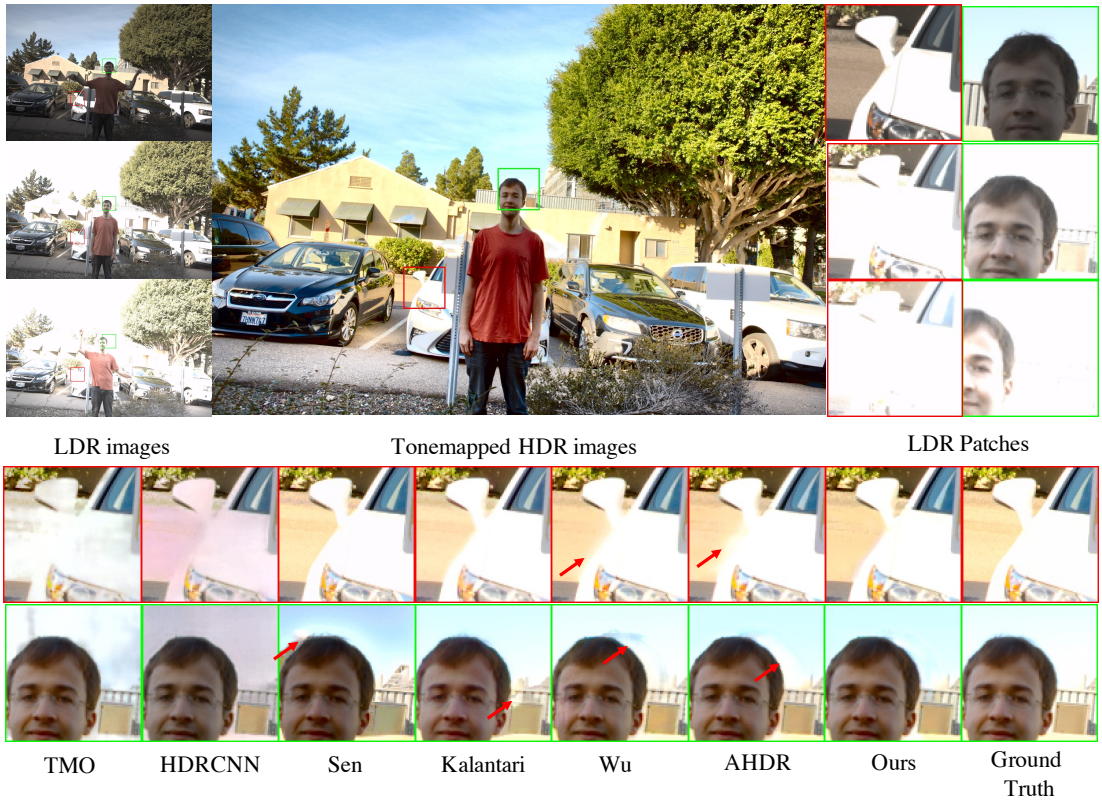


Figure 2.11: Results from the Testing data (10) of [1]. See Figure 2.6 for the explanation of the panels

Evaluation on Datasets w/o Ground Truth

We also show qualitative comparisons using Sen's [2] and Tursun's [3] datasets, which do not have ground truths. The results are shown in Figure 2.7, Figure 2.8 and Figure 2.14. The single image methods, TMO [61] and HDRCNN [60], fail to recover a sharp image and suffer from color distortion and noise. The patch based method (Sen et al [2]) produces severe artifacts in the saturated area and ghosting artifacts in the areas undergoing large motion. The same is true for the method of Kalantari et al. [1]; it produces artifacts in the areas undergoing large motion and fails to recover the details of the saturated areas. These are arguably because of the possible misalignment of optical flows and the limitation of the merging method. The results of Wu et al.'s method [5] tend to show over-smoothness and yield ghosting artifacts on the large motion areas. AHDR [54] yields artifacts in the saturated areas

(a) PSFNet (b) Ground Truth

Figure 2.12: Results of the test set of [1].

(a) PSFNet (b) Ground Truth

Figure 2.13: Results of the test set of [1].

Figure 2.14: Results for an image from the dataset of [3]. See Figure 2.6 for the explanation of the panels.

and also suffers from ghosting artifacts due to large motion. On the other hand, our method produces good results with noticeably reduced geometric and color distortions compared with others.

Table 2.2: Comparison of fusion methods on the test set of [1].

Fusion Methods	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L
Summation	43.9789	41.4092	0.9904	0.9866
Concatenation	43.9560	41.4579	0.9907	0.9867
SFFB	44.0613	41.5736	0.9907	0.9867

2.4.4 Ablation Study

We conducted experiments to evaluate the components of PSFNet. First, we compare different methods for feature fusion in the PSFB. To be specific, we replace the SFFB with concatenation or summation and evaluate the performance. Table 2.2 shows the results. It is observed that the SFFB yields better PSNR values, although there is little difference in SSIM values.

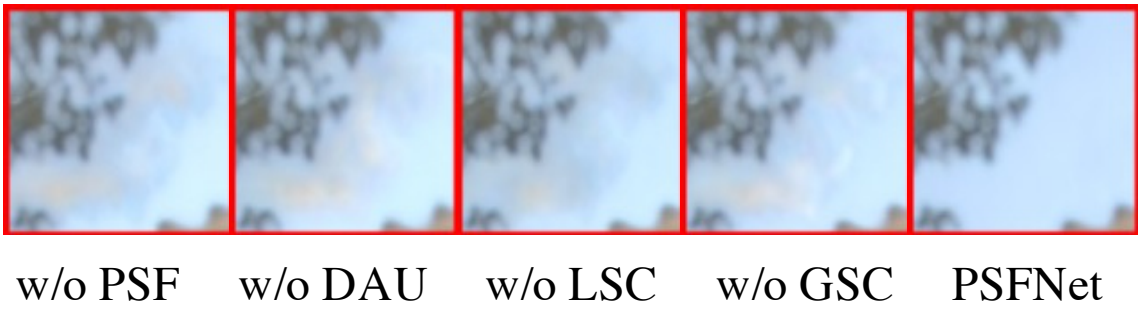


Figure 2.15: Results obtained by ablated networks.

Table 2.3: Ablation study using the test set of [1].

Methods	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L
w/o PSFBs	43.3586	40.9648	0.9902	0.9864
w/o DAB	43.7917	41.0672	0.9903	0.9856
w/o LSC	43.9923	41.4289	0.9906	0.9864
w/o GSC	44.0125	41.3756	0.9907	0.9866
PSFNet	44.0613	41.5736	0.9907	0.9867

We also evaluate individual components in PSFNet. We eliminate either the stack of PSFBs, the local skip connection (LCS), or the global skip connection (GSC) from the feature fusion network. When we eliminate the PSFB stack, we use the feature maps F_i's instead of O_i's. We also ablate the DAB from the reconstruction network. Table 2.3 shows the results. It is seen that PSFBs and DAB are essential to achieve the best performance and the skip connections (LSC and GSC) show modest

Table 2.4: Number of PSFBs used in the feature fusion network.

Number of Blocks	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L
4	43.9610	41.1000	0.9907	0.9865
5	43.9895	41.3123	0.9907	0.9865
6	44.0613	41.5736	0.9907	0.9867
7	44.0504	41.5091	0.9906	0.9867
8	43.9737	41.3107	0.9906	0.9860

contributions. Figure 2.15 shows examples of zoomed-in patches of an HDR image produced by the ablated networks. It is seen that color distortion emerges except the PSFNet (with full components).

Finally, we examine how the number of PSFBs in the feature fusion network affects the performance. Table 2.4 shows the results. It is seen that there is a peak around 6 and 7 blocks. We conclude that stacking multiple PSABs does contribute to better performance and too large a number of blocks does not lead to a good result.

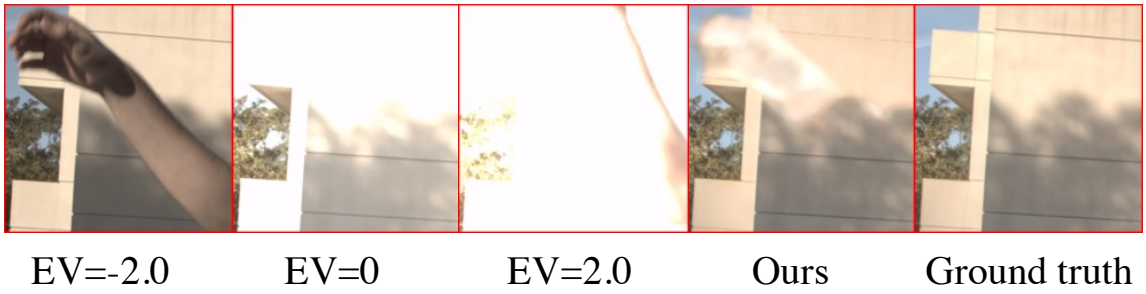


Figure 2.16: An example failure case of the proposed method.

2.4.5 Limitation

Although our method achieves good performance, as reported above, there are several cases that it cannot handle well. Figure 2.16 shows an example, where the generated image contains artifacts on the region occluded by the hand in one of the input images. The artifacts emerge because the other images do not provide useful

information about the occluded area due to overexposure. Our method does not work well for such cases; they might be better formulated as image inpainting.

2.5 Summary and Conclusion

In this section, we have proposed a new method for generating an HDR image of a dynamic scene from its LDR images. When employing deep learning, the problem reduces to first fusing the features of the input images and then reconstructing an HDR image from the fused feature. The first step of feature fusion plays a central role in generating good quality HDR images. Considering the complexity and difficulty with it, we proposed a network named PSFNet that gradually fuses the image features in multiple steps with a stack of computational blocks and the design of the component block that can effectively perform the two operations fundamental to the HDR image generation, i.e., comparing and selecting appropriate images/regions. The experimental results have validated the effectiveness of the proposed approach.

Chapter 3

Learning Regularized Multi-Scale Feature Flow for High Dynamic Range Imaging

3.1 Introduction

High dynamic range (HDR) imaging is a method to generate a larger dynamic range of illumination than standard imaging systems. It has been applied to movies [46] and computer rendering [47] to gain more information and better visual experience. As cameras that can capture HDR images are generally expensive, an alternative way to get HDR images is to reconstruct HDR images from a series of low dynamic range (LDR) images captured by a standard camera with different exposure settings. While they can reconstruct high-quality HDR images for static scenes, the existing methods tend to yield images with many ghosting artifacts for dynamic scenes, in which the imaging scenes are static captured by a hand-held camera or there are some moving objects. Increasing efforts have been invested in exploring how to remove ghosting artifacts in the multi-exposure-based HDR reconstruction. There are several methods that attempt to detect motion regions in the input LDR images and then remove these regions in the step of merging the images [48–50]. However, they tend to work well only when the motion in the input images is relatively small. When there are large motions, a large number of image pixels need to be removed,

which results in incorrect reconstruction because the information about these pixels is lost.

Another approach has also been studied, which is to align input LDR images to a reference view, and then merge them altogether for HDR image reconstruction [51–53]. Many recent methods employ convolutional neural network (CNN) to improve reconstruction image quality. However, there is still room for improvements in the area of ghosting artifacts. End-to-end learning-based approaches such as [5, 54, 73] without explicit alignment directly feed LDR images into a network to reconstruct HDR images, failing to deal with scenarios with complex motion or large disparity. As shown in Fig. 3.1, these end-to-end leaning-based methods fails to deal with the motion region. The method in [1] performs optical flow-based image alignment followed by a convolutional neural network at the merging process. Aligning images in the pixel domain is often prone to noisy or saturated pixels-induced misalignment, which leads to visible artifacts in the final synthesized presentation. In addition, the classical optical flow methods and the optical flow models pre-trained on other datasets can not deal with the occlusion region in which the ghosting artifacts often occur. As suggested in [74], feature warping can achieve better performance compared with warping the image. The method in [62] performs alignment in feature domain by using deformable convolution layers [63]. However, it has a limitation in finding long-distance correspondence; as argued in [75], deformable convolution could also lead to an unstable training process and limited generalization. Inspired by non-local structure [76], Choi et al. [77] proposed to calculate the inter-similarity between LDR images for every pixel, which are used to align non-reference features toward the reference feature. However, this non-local structure-based operation is computationally expensive and needs large memory when the size of input images is large, while the images for HDR imaging often have a large size. According to [78], Task-Oriented flow learns to handle occlusions well, though its estimated motion field differs from the ground truth optical flow. Considered the benefit from Task-Orient flow, Kalantari et al. [79] proposed a Task-Oriented flow network which is specifically designed for HDR video reconstruction and is only based on the loss for HDR video reconstruction. This Task-Oriented flow network performs better than pre-trained or classical optimization-based optical flow methods since it can deal with the occlusion,

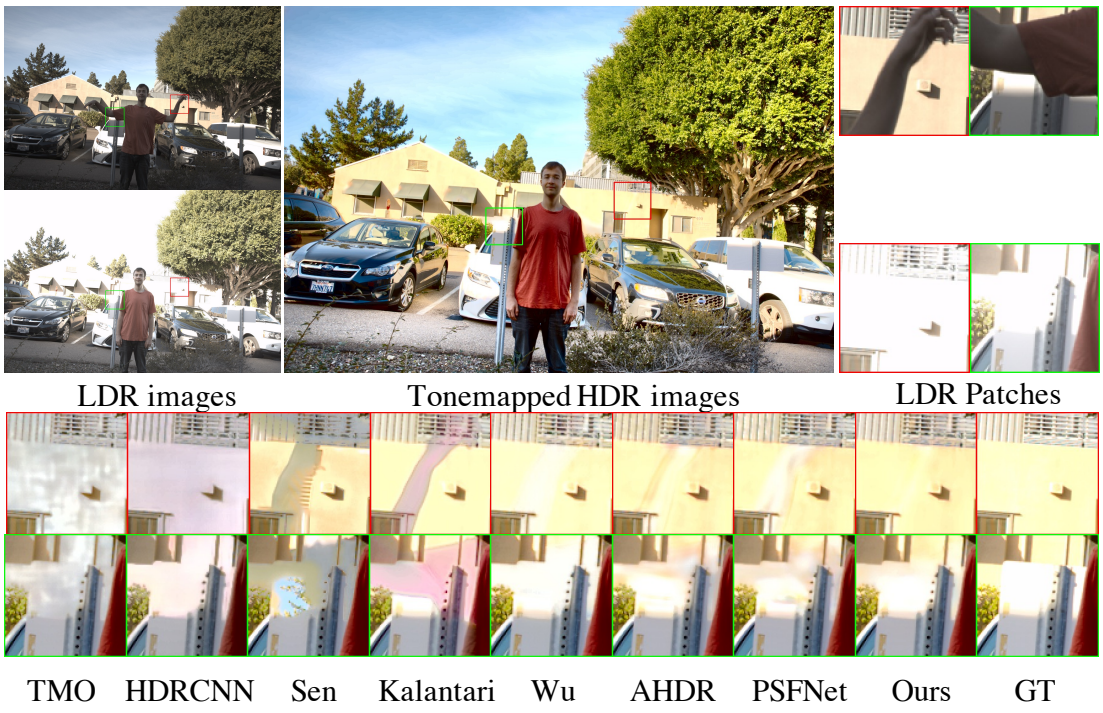


Figure 3.1: Examples of generated HDR images from the test set of [1]. The zoomed regions of different methods are highlighted.

which reduces the artifacts in occlusion regions. However, only trained on the task-specific loss (i.e. HDR reconstruction loss), the Task-Oriented flow will fail on the large saturated areas in which there are a few details. In this case, this misalignment leads to artifacts on the over-exposed regions in the reference image.

Inspired by the photometric loss [80] for self-supervised learning of optical flow, we proposed the regularized loss to provide supervision for flow learning to address the misalignment in Task-Orient flow in HDR imaging. We directly reconstruct an HDR image based on the aligned features and compute loss between this reconstructed image and the corresponding ground truth.

Differing from the previous methods [1, 79, 81] that use the existing optical flow model like SPyNet [82] and PWC-Net [83], we design a simple but effective network for learning the flow for feature alignment. We remove the context encoder in the flow network and directly use the features for HDR image reconstruction as the input for flow estimation. We argue that this flow structure can achieve better alignment performance in the HDR imaging task since there are large illumination changes and

large saturated areas in image space, while the extracted features for HDR reconstruction can provide rich information to avoid misalignment. And we name this flow structure as feature flow.

To this end, we propose a new network that enables end-to-end training, including alignment. The proposed method consists of two networks: *alignment network* and *merging network* in this order. The alignment network extracts multi-scale features from the input LDR images and estimates optical flow. It then aligns the non-reference LDR images to the reference LDR image in feature space using the estimated flow. The merging network takes the aligned features and multi-scale features as input and generates a final HDR image using a residual attention mechanism. Experimental results show that our method can achieve better quantitative and qualitative evaluation performance than the existing state-of-the-art methods on the commonly used public test datasets.

3.2 Pioneering Works

3.2.1 Motion Removal based Methods

These methods is firstly to detect the motion region and then remove these pixels on the motion region in the merging processing. Khan et al. [48] use a non-parametric model to compute weights iteratively and apply these computed weights to pixels to fuse multiple LDR images to obtain final HDR images. Heo et al. [49] detect motion regions based on the joint probability densities and refine these regions by using energy minimization. Jacobs et al. [56] propose a method to detect moving pixels based on the difference between the LDR images. Zhang et al. [57] propose motion detection method based on the image gradients between different images. Lee et al. [58] considered that the noise, moving objects, and distortions as outliers, so they proposed a low-rank model to reconstruct HDR images. Following their method, Yan et al. [50] proposed a sparse model to detect motion regions. When the motion in LDR images is small, motion-removal-based methods can achieve satisfactory results. However, when the motion is large, a large number of pixels are unavoidably removed in the merging stage, causing undesirable artifacts in the generated HDR images.

3.2.2 Alignment based Methods

Most alignment based methods adopt optical flow and its variants to align LDR images and then merge aligned LDR images to generate corresponding HDR images. Bogoni [51] use optical flow to estimate motion filed between LDR images and then warped and aligned these LDR images by using the computed motion field. Instead of fusing LDR images in the spatial domain, Kang et al. [59] firstly utilize the information of exposure time and converted LDR images into luminance domain. In the fusion process, a method was proposed to eliminate artifacts by using the optical flow. Sen et al. [2] propose a method based on a patch-matching algorithm for HDR reconstruction. Hu et al. [52] propose a displacement estimation method which converts images by the intensity mapping function and then merging images in the transformed domain for HDR image generation, which implicitly align LDR images by searching and aggregating similar patches. Hafner et al. [53] propose a method to jointly estimate the optical flow and reconstruct HDR image. However, since the alignment process in the image domain is vulnerable to large motion and excessively dark or bright regions, these methods tend to generate artifacts in the aligned images.

3.2.3 CNN Based Methods

As with other computer vision tasks, CNNs have been applied to HDR imaging. Eilertsen et al. [60] propose an encoder-decoder network to generate an HDR image from a single LDR image. Endo et al. [61] synthesize multiple LDR images with different exposures from a single LDR image by CNNs, and then merge them to reconstruct an HDR image. These single-image-based methods are unable to reconstruct the textures on saturated regions accurately.

To generating more accurate images, more attention is paid on obtaining HDR images from a sequence of LDR images captured with different exposures. Kalantari et al. [1] propose the first CNN-based method for HDR imaging, where the input LDR images are first aligned by optical flow and then the aligned LDR images are fed to CNNs to reconstruct an HDR image. In stead of using explicit alignment, Wu et al. [5] directly concantenated the features extracted from input LDR images and forwarded them to a deep model with the U-net structure to reconstruct HDR images. Yan et

Figure 3.2: Three LDR images of the same scene captured with three different exposures. L_1, L_2, and L_3 denote the images captured with the low, medium, and high exposure, respectively.

al. [55] introduce a non-local structure [76] into the U-net as for implicit alignment. Yan et al. [54] propose an attention module to learn to identify misaligned elements before merging the LDR images. Pu et al. applied the deformable convolution [63] to multi-scale features, which aligned LDR images in a pyramidal manner, and reconstructed the corresponding HDR images. To reduce the computational cost by CNN-based methods, Prabhakar et al. [84] propose an efficient method that performs all operations in low resolution and upscales the result to the required full resolution. Similar to our work, a few studies consider using optical flow for the alignment. But they either use a pre-trained estimator [81] or optimize the estimator through the reconstruction loss [79], which may lead to suboptimal results.

3.3 Proposed Method

Given a series of LDR images, $L_1, L_2, ..., L_k$, captured with different exposures, the goal of HDR imaging is to generate an HDR image H corresponding to a selected reference image L_r.

There are two samples with different exposure settings shown in Fig. 3.2. The sample in the first row shows that L_3 has little effect for image restoration of L_2

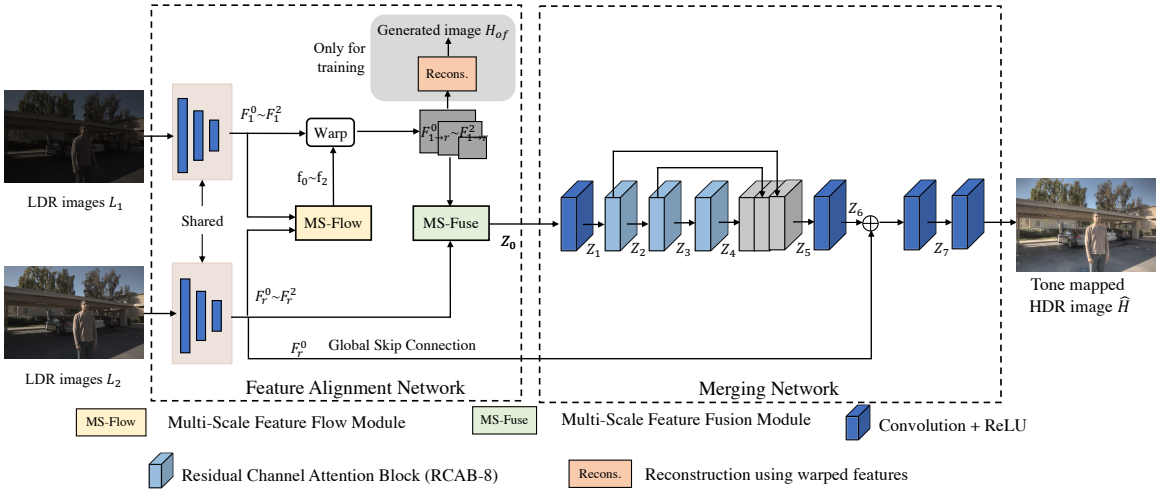


Figure 3.3: Overview of the proposed network. It consists of two sub-networks: feature alignment and merging networks. The alignment network warps the features of the non-reference images onto those of the reference image using optical flow. The merging network takes the warped features as input and reconstructs an HDR image.

since there are large areas of over-exposure region in L_3. While the sample in the second row shows that L_3 can be helpful for image restoration of L_2. In this case, the model can easily produce high-quality HDR images due to the efficient information. Without the input of L_3, the model can also generate a high-quality HDR image though the model needs to be more effective.

Unlike the previous methods taking three LDR images as input, we use two LDR images, L_1 and L_2 as inputs, sorted in the order of exposures and set L_2 as the reference image by considering the properties in L_3. And two images for input can also reduce computational costs.

Following the settings of previous studies [1, 54], we first map the LDR images into the HDR domain using gamma correction and then feed them into the network. To be specific, we map L_i to H_i by

$$H_i = L_i^{\gamma}/t_i, \quad i = 1, 2, \tag{3.1}$$

where γ denotes the gamma correction parameter and followed [64] we use $\gamma = 2.2$ in this section. t_i is the exposure time of L_i. As the suggestion in [1], we concatenate L_i and H_i in the channel dimension to obtain a six-channel tensor $X_i = [L_i, H_i], i = 1, 2,$

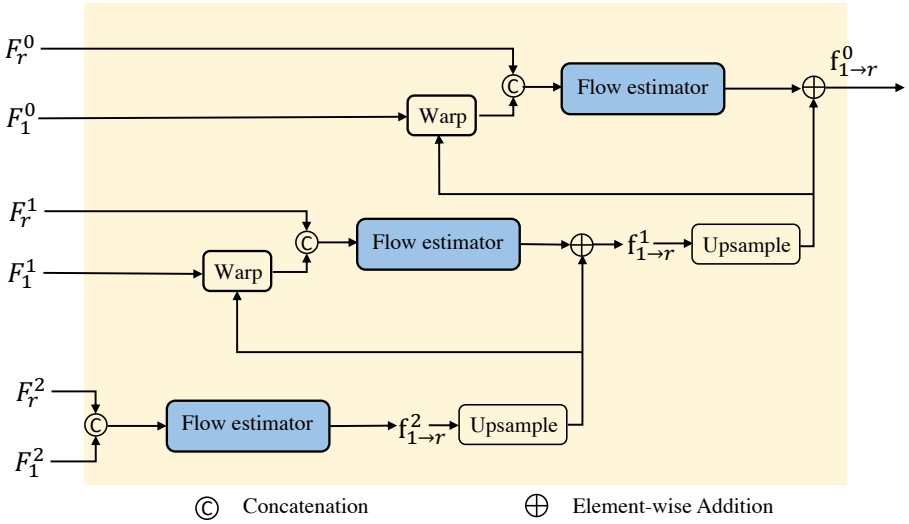


Figure 3.4: Architecture of the multi-scale feature flow module (MS-Flow). It follows the coarse-fine manner to generates multi-scale optical flows and the multi-scale feature maps aligned to the reference image.

and input X_1 and X_2 to the network.

Our network consists of two sub-networks, the alignment network and merging network, as shown in Fig.3.3. We first describe the alignment network (Sec.3.3.1) and then explain the merging network (Sec.3.3.2).

3.3.1 Feature Alignment Network

The feature alignment network first extracts multi-scale feature maps from the input tensor X_i. Specifically, the feature extractor consists of a convolution layer with stride = 1 and the following two layers with stride = 2, which forms multi-scale feature maps with scale = 0, 1, and 2. We represent the feature map at scale $s \in \{0, 1, 2\}$ as $F_i^s \in \mathbb{R}^{H_s \times W_s \times C_s}$, where $H_s = H/2^s$, $W_s = H/2^s$, and $C_s = C$. These feature maps will be used in the subsequent modules. For clarity, we use the index $r(= 2)$ to indicate the reference LDR images; thus, $X_r = X_2$ and $F_r^s = F_2^s$ in what follows.

Multi-Scale Feature Flow Module (MS-Flow)

Following SPyNet [82] and PWC-Net [83], we estimate the optical flow in a coarse-to-fine manner, as shown in Fig. 3.4. The estimated flows at the coarser scales can capture the large motions. On the other hand, the flows at the finer scales will be helpful to capture small motions.

We first concatenate the coarsest scale features F_i^s with F_r^s in the channel dimension and feed it to a flow estimator. The estimator consists of five convolution layers with 7×7 kernel size and generates the s-th scale optical flow $\mathbf{f}_{1\to r}^s$. Then, we upsample $\mathbf{f}_{1\to r}^s$ by factor $= 2$ and use it to warp the non-reference feature map F_1^{s-1} onto the reference feature map F_r^{s-1}. Specifically, we map each pixel $\mathbf{p}_1^{s-1}$ in F_1^{s-1} to its estimated correspondence in F_r^{s-1} as

$$\mathbf{p}_r^{s-1} = \mathbf{p}_1^{s-1} + \tilde{\mathbf{f}}_{1\to r}^s(\mathbf{p}_1^{s-1}), \tag{3.2}$$

where $\tilde{\mathbf{f}}_{1\to r}^s$ represents the upsampled flow of the s-th scale flow $\mathbf{f}_{1\to r}^s$. We then concatenate the warped and reference feature maps in the channel dimension and feed it to the subsequent flow estimator. The output of the flow estimator is then element-wise added to the upsampled flow, yielding the flow at scale $s-1$. We iterate this procedure for $s = 1$ and 0, obtaining multi-scale optical flows $\mathbf{f}_{1\to r}^s$ and the warped multi-scale feature maps $F_{i\to r}^s$.

Multi-Scale Feature Fusion Module (MS-Fuse)

As shown in Fig. 3.5, the multi-scale feature fusion module takes the concatenated feature maps at each scale $\bar{F}^s = [F_{1\to r}^s, F_r^s]$. We apply a convolution layer with the kernel size of 3×3 followed by ReLU to the finest feature map $\bar{F}^0$ to obtain a feature map O^0. For the feature maps $\bar{F}^1$ and $\bar{F}^2$, we first apply a convolution layer with the same kernel size and then upsample the outputs with bilinear interpolation so that the resulting maps become the same size as the finest one. Finally, all the outputs are concatenated as $Z_0 = [O^0, O^1, O^2]$ and then used as input for the following merging network.

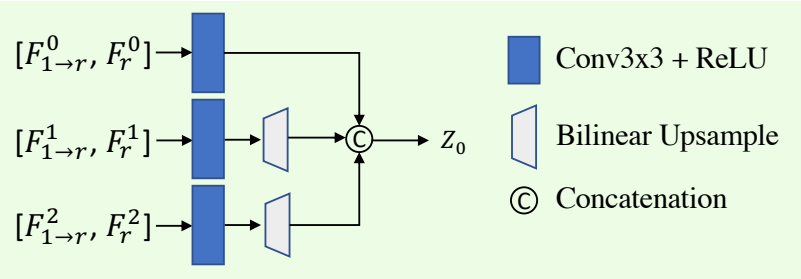


Figure 3.5: Multi-scale feature fusion.

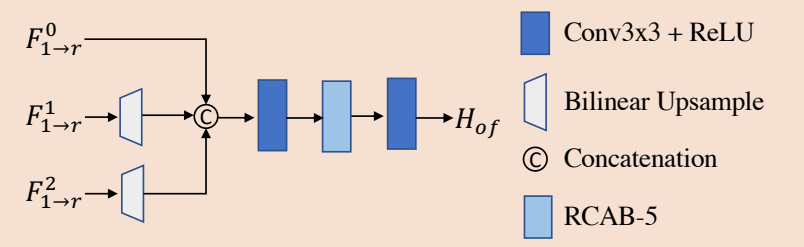


Figure 3.6: Reconstruction of H_{of}.

Reconstruction using Warped Features

Unlike the previous HDR imaging studies using optical flow, we reconstruct an HDR image H_{of} using the feature maps $F^s_{1\to r}$, which are the feature maps right after warping by the optical flow. Our intention behind this reconstruction is to directly guide the network to generate accurate optical flow and perform better alignment. As shown in Fig. 3.6, we first upsample the feature maps $F^1_{1\to r}, F^2_{1\to r}$ so as to be the same size as the finest feature map $F^0_{1\to r}$. We then concatenate them and feed them to a series of convolution layers with the kernel size of 3×3 followed by ReLU and five residual channel attention blocks (RCAB) [4]; see Fig. 3.7 for the detail of the RCAB. We then calculate ℓ_1 loss between the reconstructed HDR image H_{of} and its ground truth HDR image, as will be explained later.

3.3.2 Merging Network

Following the previous methods [4, 54], we employ an attention mechanism to merge the feature maps and generate an HDR image; in specific, we use the RCAB. As shown in Fig. 3.3, the merging network takes the concatenated feature maps $Z_0 = [O^0, O^1, O^2]$. We apply a convolution layer and three RCABs to Z_0 and then

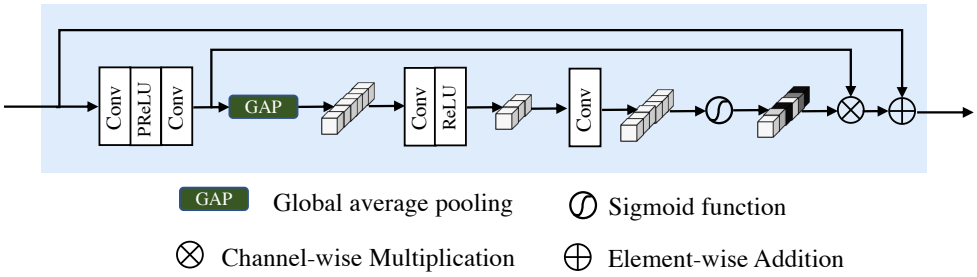


Figure 3.7: Architecture of a residual channel attention block (RCAB) [4].

concatenate the outputs of each RCAB as $Z_5 = [Z_2, Z_3, Z_4]$. Applying three convolutions and a global skip connection with F_r^0, the merging network outputs a final HDR image.

3.3.3 Loss Function

Following [1], we consider the optimization in the domain of tonemapped HDR images because the HDR images are usually displayed after tonemapping and training the network on the domain is more effective than that on the original domain of HDR images. Thus, we employ the μ-law for tone mapping as suggested in [1], which is formulated as

$$T(H) = \frac{log(1 + \mu H)}{log(1 + \mu)}, \tag{3.3}$$

where μ is set to $5,000$ throughout our experiments. It is also reported in [54,85] that minimizing the L_1 norm between the predicted HDR image $\hat{H}$ and its ground truth H in the tone-mapped domain works better than others. Following their studies, we use the following ℓ_1 loss,

$$L_{tm} = \|T(\hat{H}) - T(H)\|_1. \tag{3.4}$$

For the standard optical flow estimators such as SPyNet [82] and PWC-Net [83], they are trained on the datasets with the standard exposure settings (e.g. Sintel [86], KITTI [87], and Middlebury [88]). However, there is no dataset containing ground truths of optical flow maps for the HDR imaging task. Inspired by the photometric loss [80] for self-supervised learning of optical flow, we use ℓ_1 loss between the reconstruction H_{of} and its ground truth H in the tone-mapped domain to provide

supervision for the optical flow learning,

$$L_{reg} = \|T(H_{of}) - T(H)\|_1. \tag{3.5}$$

Our total loss is taken as the weighted sum of two losses

$$L = L_{tm} + \lambda L_{reg}, \tag{3.6}$$

where we use $\lambda = 2$ in this section.

3.4 Experiments

3.4.1 Experimental Settings

Training Data

To train our network, we adopt the HDR dataset [1] which consists of 74 samples for training and 15 samples for testing. We use the former for training our model. Each sample includes a ground truth HDR image and three LDR images with different exposure settings of $\{-2, 0, +2\}$ or $\{-3, 0, +3\}$. All the images are resized to the resolution of 1000×1500.

Testing Data

Following recent studies, we choose the following datasets for testing. We evaluate our method on the 15 scenes of the dataset of [1], where we perform a quantitative evaluation using the provided ground truths. We also test the proposed method on the datasets of Sen et al. [2] and Tursun et al. [3]. Since these datasets do not contain ground truths of HDR images, we compare the reconstructed HDR images by our method with those by state-of-the-art methods for qualitative evaluation.

Evaluation Metrics

As used in the existing studies, we use PSNR-μ and SSIM-μ for primary metrics, which are PSNR and SSIM values in the tone-mapped domain using μ-law. We show

PSNR and SSIM values in the linear domain, which are denoted by PSNR-L and SSIM-L for completeness. We also report HDR-VDP-2 [70], which is designed to evaluate the quality of HDR images.

3.4.2 Implementation Details

For training, we first crop the training images into patches of 256×256 pixel size with a stride of 128 pixels. We then apply random rotation and flipping for data augmentation to avoid over-fitting. We use the Adam optimizer [45] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, initial learning rate 1×10^{-4} and set the batch size to 8. We train our model for 210 epochs and employ the cosine annealing strategy [71] to steadily decrease the learning rate from an initial value to 1×10^{-6}. We implement our model using PyTorch [72] on NVIDIA GeForce RTX 2080 GPUs.

3.4.3 Comparison with the State-of-Art Methods

We compare the proposed method with existing methods. Specifically, we compare our model with two HDR imaging methods based on a single LDR image, TMO [61] and HDRCNN [60], and five HDR imaging methods based on multi LDR images, the patch-based method [2], the flow-based method with CNN merger [1], the U-net structure without optical flow [5], the attention-guide method (AHDR) [54], pyramidal alignment network (PANet) [62], and progressive and selective fusion network (PSFNet) [73]. For all the methods, we used the authors' code for comparison, except for [62] since their code is not available as of the time of writing this paper.

Evaluation on Kalantari et al.'s Dataset

Figure 3.8 and 3.9 show two examples on the test set of [1]. The input LDR images contain saturated background and foreground motions. We can observe from the results of the single-image methods, TMO [61] and HDRCNN [60], that they cannot sufficiently recover the detailed textures and generate artifacts in the over-exposed regions; they also suffer from the color distortion. The patch-based method of Sen et al. [2] generates some artifacts due to the failure of finding patches correctly. Kalantari et al.'s method [1] cannot completely eliminate the effects of the occlusion.

Table 3.1: Quantitative comparison on the Kalantari's test sets [1]. The numbers in the table are the average values of the 15 test images.

Methods	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L	HDR-VDP-2
TMO [61]	8.3120	8.8459	0.5029	0.0924	44.3345
HDRCNN [60]	13.7054	13.8956	0.5924	0.3456	47.5690
Sen [2]	40.9689	38.3425	0.9859	0.9764	60.3463
Kalantari [1]	42.7177	41.2200	0.9889	0.9829	61.3139
Wu [5]	41.9977	<u>41.6593</u>	0.9878	0.9860	61.7981
AHDR [54]	43.7013	41.1782	0.9905	0.9857	62.0521
PANet [62]	43.8487	41.6452	0.9906	<u>0.9870</u>	62.5495
PSFNet [73]	<u>44.0613</u>	41.5736	<u>0.9907</u>	0.9867	**63.1550**
Ours	**44.3298**	**41.8936**	**0.9911**	**0.9878**	<u>63.1190</u>

The method of Wu et al. [5] cannot deal with over-exposed regions and then produces artifacts on motion areas. Non-aligned methods (i.e. AHDR [54] and PSFNet [73]) yield artifacts in the saturated areas and also suffer from ghosting artifacts due to the large motions. Compared with them, our proposed method produces less color distortion and recovers the textures more accurately, leading to the best qualitative results.

Table 3.1 shows the quantitative evaluation on the same dataset. In specific, we report the averaged values over 15 test scenes. It can be seen that the proposed method achieves better performance than the others in terms of PSNR-μ, SSIM-μ, PSNR-L and SSIM-L. Also, our method achieves comparable performance to the state-of-the-art method [73] in terms of the HDR-VDP-2 metric.

We also compute PSNR-μ and PSNR-L on different masks. As shown in 3.12, there are two kinds of masks. Mask1 can be considered as the occluded region while Mask2 can be considered as the combination between the occlusion region and the object. As shown in Table 3.2, Wu et al.'s method [5] achieves the best performance in Mask1 setting while our method achieves the best performance in Mask2 setting. But our method can produce much better visualization results than Wu et al. [5] as shown in 3.13. The potential explanation about the high numerical result achieved

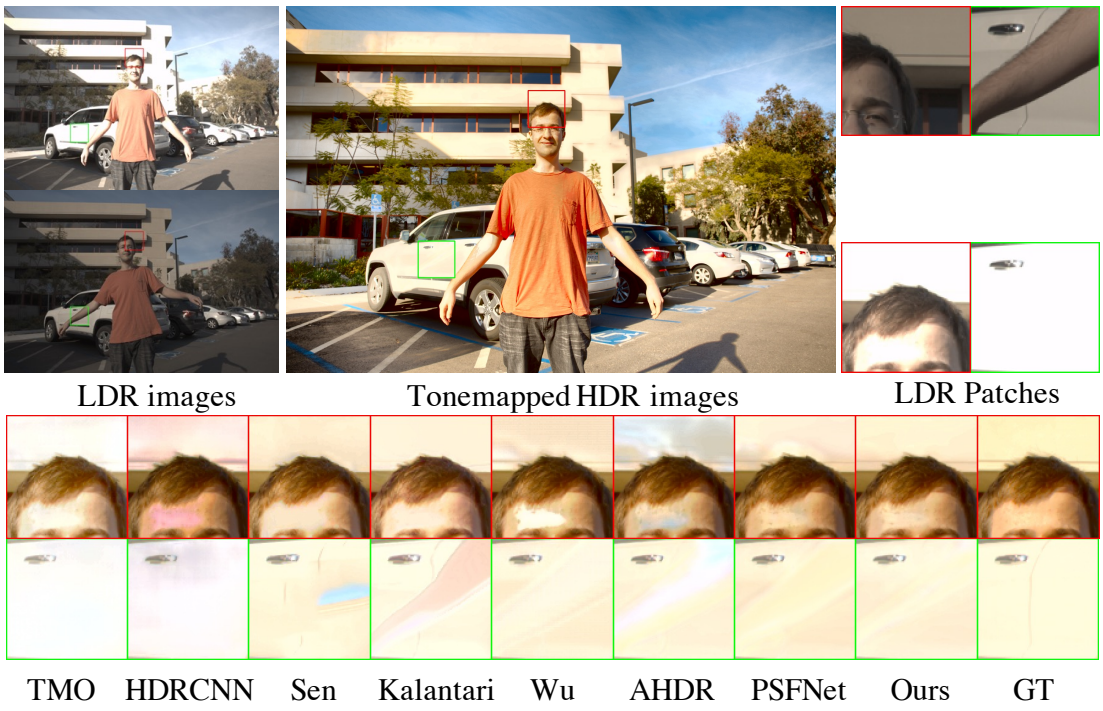


Figure 3.8: Results from the Testing data (08) of [1]. Upper row from left to right: the two input LDR images, the HDR image produced by the proposed method, and (zoomed-in) LDR image patches with two identical positions/sizes (in green and red). Lower row: the same patches of the HDR images produced by different existing methods

by Wu et al. [5] on Table 3.2 is that the most artifacts produced by Wu et al. [5] are color distortion and the over-exposure region, which may not have large numerical error.

Evaluation on Datasets w/o Ground Truth

We also provide comparisons using Sen's [2] and Tursun's [3] datasets. These datasets do not have ground truths of HDR images and thus we qualitatively compare the generated HDR images.

Some examples of the results are shown in Fig. 3.10 and 3.11. The single image methods, TMO [61] and HDRCNN [60], generate serious noises and color distortions in the under-exposed regions. The patch-based method (Sen et al [2]) also generates severe artifacts. The method of Kalantari et al. [1] produces artifacts due to the align-

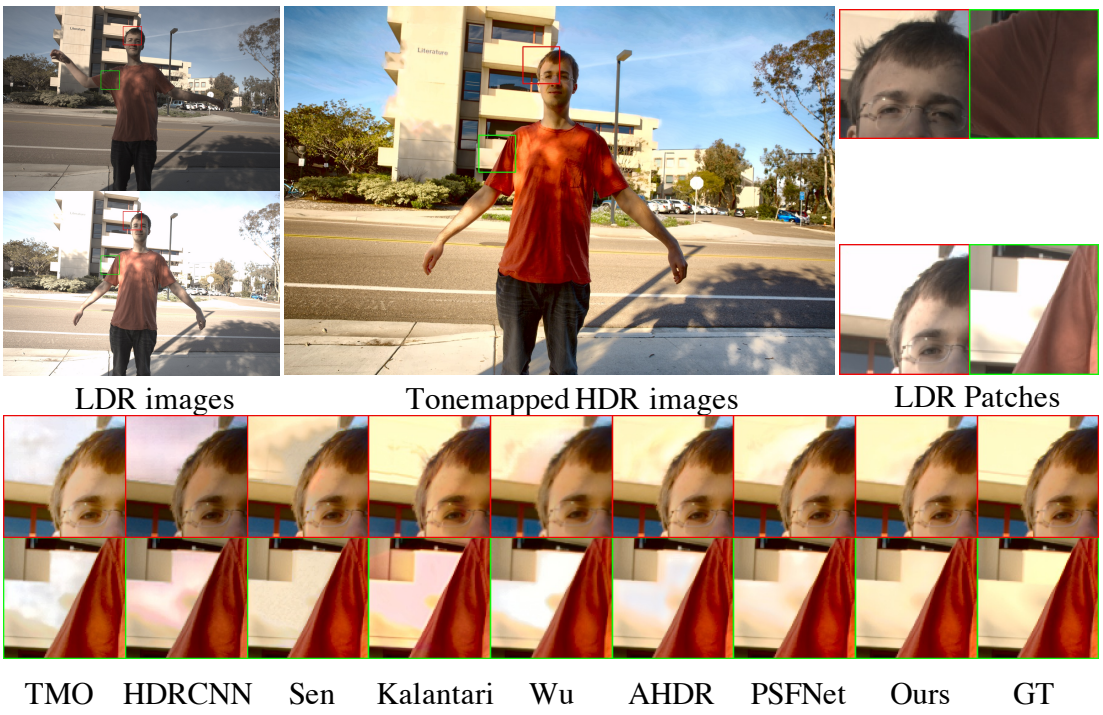


Figure 3.9: Results from the Testing data (09) of [1]. See Figure 3.8 for the explanation of the panels

ment error and also generates serious noises in the under-exposed regions. These are arguable because of the misalignment by the estimated optical flow and the limitation of the merging method. Wu et al.'s method [5] tends to yield over-smoothness and generate ghosting artifacts on the large motion areas. AHDR [54] yields color distortions and also suffers from ghosting artifacts due to large motions shown in Fig. 3.11. PSFNet [73] generates ghosting artifacts in the motion regions and generates the geometric distortions shown in Fig. 3.10. On the other hand, our method produces better results with noticeably reduced geometric and color distortions compared with others.

3.4.4 Ablation Study

We demonstrate the effectiveness of each component in the proposed method. We use the same configurations as those used above unless otherwise noted.

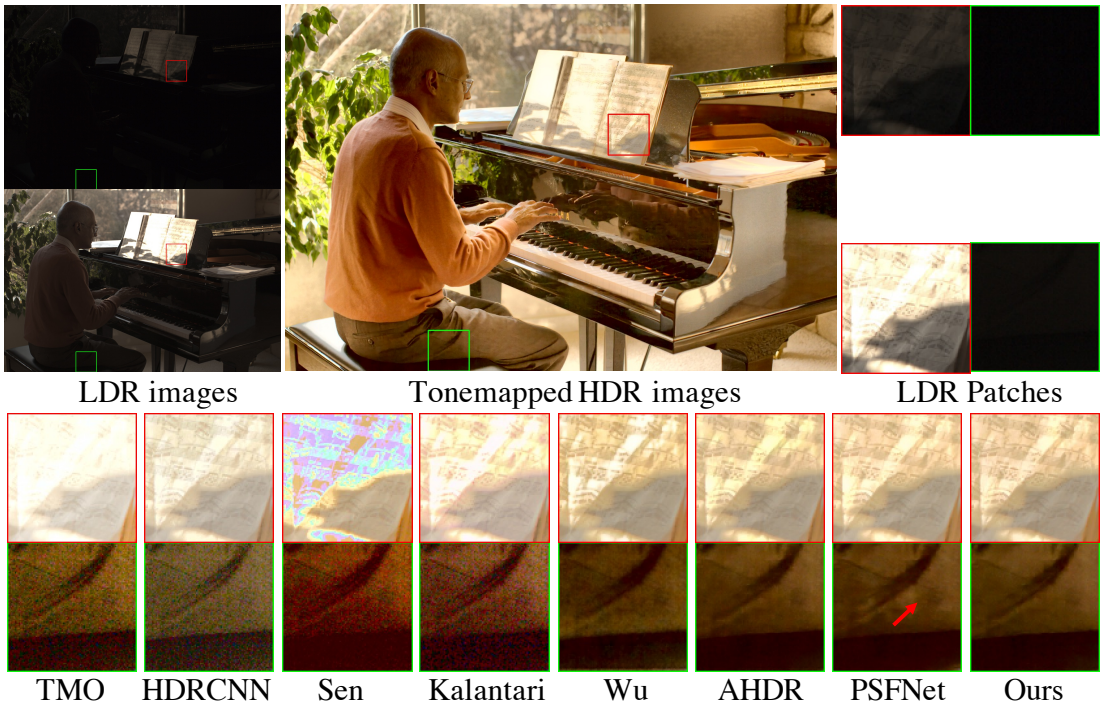


Figure 3.10: Example of Sen et al.'s dataset [2]

Effects of Optical Flow Learning

First, we verify the effect of the proposed loss in Eq. 3.5 by changing the value of the parameter λ. When $\lambda = 0$, the network will be trained only using the ℓ_1 loss (i.e. Eq. 3.4) between the final outputs and their ground truths. This is equivalent to the previous methodology using optical flow [79]. It should be noted that although the work [79] tackles the HDR video reconstruction, the approach also works on the HDR imaging task. As shown in Fig. 3.14, the model without the proposed loss (i.e. $\lambda = 0$) generates sever color and geometry distortions. In contrast, the models with the proposed training (i.e. $\lambda > 0$) significantly improve the reconstruction results. We also quantitatively evaluate them on the Kalantari et al.'s test set [1]. As shown in Table 3.3, our proposed training with $\lambda = 2$ achieves the best performance. Even though the gain by the proposed training is not so large, the artifacts usually appear in a small area of an image, and they have only small impacts on these metrics.

To further verify the effectiveness of the regularization loss, we show the image warped by the flow learning by different λ settings. When $\lambda = 0$, our model is only

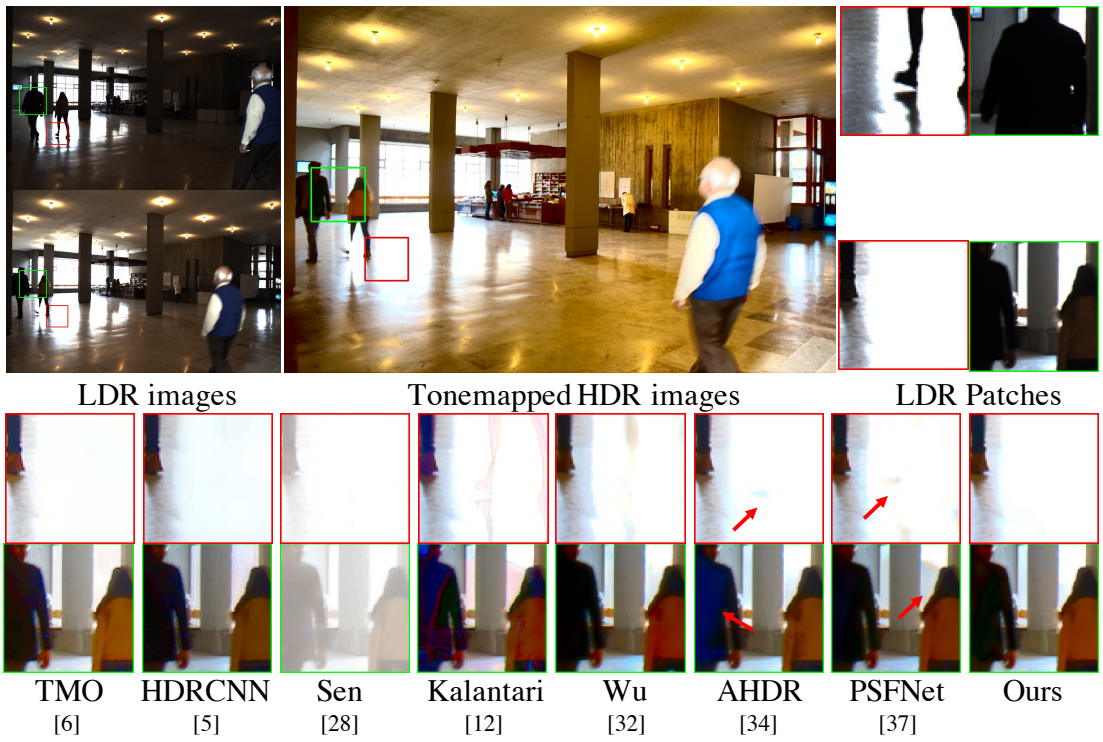


Figure 3.11: Example of Tursun et al.'s dataset [3]

optimized by the L_{tm} in Eq. 3.4. As shown in Fig. 3.15, this optimization fills the occluded region by the hand somewhat better than that with the L_{reg} ($\lambda = 2$), but it produces severe distortions in some regions.

The reconstruction of H_{of} is only based on the warped features from the non-reference images. Then L_{reg} computed on H_{of} and the ground truth imposes a heavy constraint on these warped features to provide more accurate gradients to the warping field than the L_{tm} which involves both the warped features and the features from the reference image. As shown in the Fig. 3.15 image warped by the flow when $\lambda = 2$, the occlusion can be partially addressed since there are some mask regions that cannot be filled, but no extra severe distortion is introduced.

Effects of Different Configurations

Since our method has several design choices, we conduct experiments to examine which configuration shows the best. Specifically, we examine the effect of channel attention (CA), multi-scale feature flow module (MS-Flow), feature flow module (FF),

Table 3.2: Quantitative results on different masked regions in the Kalantari's test sets [1].

	Mask1		Mask2	
Methods	PSNR-μ	PSNR-L	PSNR-μ	PSNR-L
TMO [61]	13.9311	8.6773	10.8671	8.9529
HDRCNN [60]	17.0306	9.5235	15.2617	12.072
Sen [2]	23.6155	11.0598	29.1973	17.0423
Kalantari [1]	26.9829	13.4805	32.1771	19.656
Wu [5]	**33.9415**	**20.4405**	38.1358	**25.3188**
AHDR [54]	32.3999	17.9917	37.3266	23.8053
PSFNet [73]	32.7707	18.2808	37.3645	23.9271
Ours	33.0874	18.754	**38.2312**	24.7554

(a) L_1 (b) L_2

(c) Mask1 (d) Mask2

Figure 3.12: Two different masks on Testing data (08) in [1].

(a) Wu (b) Ours

Figure 3.13: Visualization comparison between Wu et al. [5] and ours on dynamic scenes in [1].

and multi-scale feature (MS-Fuse). The results are shown in Table 3.4. When we eliminate the MS-Flow, we use a single convolution layer to extract feature maps and

Table 3.3: Results obtained with different λ values in Eq. 3.6.

	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L
λ=0	44.1229	41.8947	0.9911	0.9875
λ=1	44.0978	41.7622	0.9910	0.9868
λ=2	44.3298	41.8936	0.9911	0.9878
λ=3	44.1238	41.7043	0.9911	0.9871

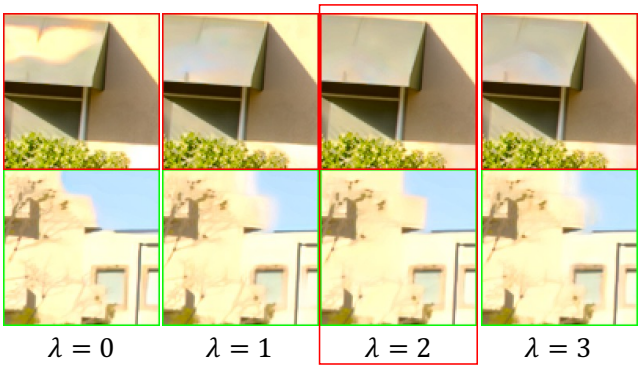


Figure 3.14: Results obtained with different λ values in Eq. 3.6. It can be seen that our proposed training (i.e. $\lambda > 0$) significantly improve the reconstruction results.

then concatenate them as Z_0. We also do not reconstruct HDR images using the warped feature maps. When we eliminate the FF, we also do not reconstruct HDR images and directly use F_1^s as $F_{1\to r}^s$ $(s = 1, 2, 3)$ since there is no optical flow available for the alignment. We can observe from Table 3.4 that CA and MS-Flow are essential to achieve better performance. Figure 3.16 shows some examples of the zoomed-in patches produced by the models with different configurations. It can be seen that artifacts appear except the MSFFNet (with full components).

We compare the proposed multi-scale feature flow estimation module with other optical flow methods (SPyNet [82] and PWC-Net [83]). We replace the FF with the optical flow (SPyNet and PWC-Net) in the proposed network named MSFFNet w/o FF w/ SPyNet and MSFFNet w/o FF w/ PWC-Net. As shown in Table 3.4, the proposed method achieves better performance than other optical flow-based methods even the PWC-Net has a more complicated network structure (e.g. with correlation layer [89]) than ours. Since our feature flow network can not be pre-trained on other

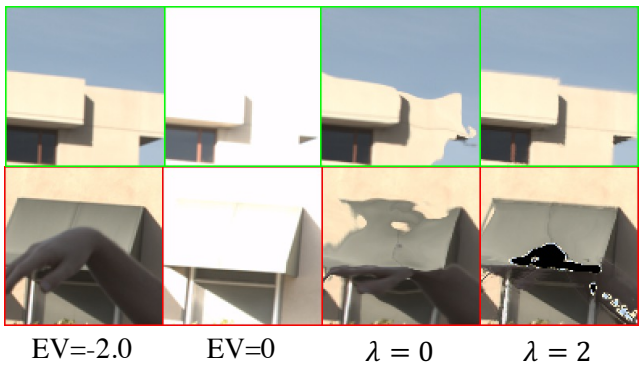


Figure 3.15: An example of the low exposure images warped by the flow for different λ values. Image from left to right are the low and medium exposure image, low exposure image warped by the flow when $\lambda = 0$, low exposure image warped by the flow when $\lambda = 2$.

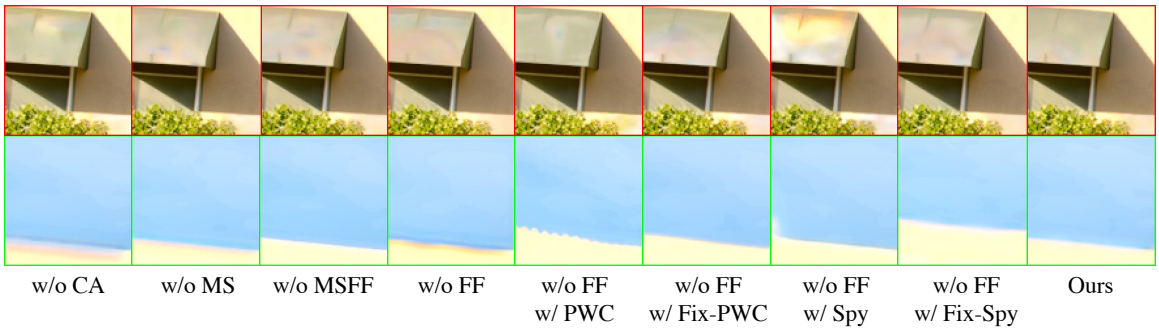


Figure 3.16: Results obtained by ablated networks.

datasets with optical flow ground truth, we also compare the pre-trained optical flow network named MSFFNet w/o FF w/ fixed-pre-trained SPyNet and MSFFNet w/o FF w/ fixed-pre-trained PWC-Net with the model trained on the proposed regularized loss to demonstrate the effectiveness of the proposed regularized loss. These two ablated networks are trained by only using tone-mapped loss in Eq. 3.4. As shown in Table 3.4, the model trained on the proposed regularized loss (MSFFNet w/o FF w/ SPyNet and MSFFNet w/o FF w/ PWC-Net) achieve better performance than the pre-trained model (MSFFNet w/o FF w/ fixed-pre-trained SPyNet and MSFFNet w/o FF w/ fixed-pre-trained PWC-Net). As shown in Figure 3.16, there is severe color and geometry distortion in the images generated from the optical flow-based method.

Table 3.4: Results of ablation tests on Kalantari's test set. The upper row shows the effects of channel attention (CA), multi-scale feature flow module (MS-Flow), feature flow module (FF), and multi-scale feature fusion module (MS-Fuse). The lower row shows the effects of the choice of optical flow.

Methods	PSNR-μ	PSNR-L	SSIM-μ	SSIM-L
MSFFNet w/o CA	44.0377	41.8150	0.9909	0.9875
MSFFNet w/o MS	44.1236	41.8216	0.9908	0.9869
MSFFNet w/o FF	43.9466	41.3520	0.9909	0.9867
MSFFNet w/o MSFF	43.6752	41.4698	0.9908	0.9868
MSFFNet w/o FF w/ SPyNet	43.9717	41.3563	0.9905	0.9852
MSFFNet w/o FF w/ fixed-pre-trained SPyNet	43.6611	41.6913	0.9896	0.9821
MSFFNet w/o FF w/ PWC-Net	44.1436	42.0084	0.9911	0.9879
MSFFNet w/o FF w/ fixed-pre-trained PWC-Net	43.3769	41.5546	0.9891	0.9808
MSFFNet	44.3298	41.8936	0.9911	0.9878

3.5 Summery and Conclusion

in this section, we propose a new method for generating an HDR image of a dynamic scene from its LDR images along the alignment-before-merging direction. The first step of feature alignment plays a central role in generating high-quality HDR images. Trained by the regularized loss, the multi-scale feature flow module can effectively learn the flow for alignment even in occlusion regions and the large saturated areas, which greatly reduce the artifacts in these regions. After the alignment by the estimated flow, the features from the non-reference image will be fused with the features from the reference image to reconstruct an HDR image. The experimental results have validated the effectiveness of the proposed approach.

Chapter 4

Defocus Map Guided Network for Defocus Deblurring

4.1 Introduction

Defocus blur is inevitable when the rays from a scene not lying on the focal plane of the camera converge to a region rather than a point on the image plane and the region is called the circle of confusion (COC) [26]. Using a large aperture allows more light to pass through the lens in a shorter exposure time. But this results in a shallow depth of field (DoF), thereby causing defocus blur. Shallow DOF is useful to make the subject stand out from the blurry background and foreground. But on the other hand, defocus blur causes visual information losses, which is important for other tasks like image understanding. Thus, recovering the blurry images can help improve the performance on these tasks.

However, defocus deblurring is still a challenging task due to the spatial variant blur, i.e., the level of blur for each pixel is different. For example, the scene on the focal plane is captured sharply while the scene out of the focal plane is captured in blurry. And the level of defocus is usually depicted by the point spread function (PSF), resulting in a pixel-wise defocus map.

The defocus blur can be modeled by

$$I_b = K * I_c + N \tag{4.1}$$

where I_b is the blurry image, I_c is the clean image, K is the blur kernel and N is the additive noise [90]. Thus the natural idea for defocus deblurring is to follow two steps to address this problem, i.e., first to estimate the defocus map indicating the level of defocus blur, then apply non-blind deconvolution [27–29]. These classical methods can achieve satisfying results under the low or medium level defocus blur but it is hard for these methods to achieve sharp results under high level blur. Some image priors are utilized to improve the performance but these priors works well under some particular scenes but may fail to cover the real-world scenes. What is more, the classical methods need iterative optimization to achieve sharp results, which is time-consuming.

In the era of deep learning, many CNN-based based approaches are proposed for deblurring problem. Nah et al. [91] propose multi-scale structure for dynamic scene deblurring and achieve good visual results. Kupyn et al. [92, 93] utilize the generative adversarial networks (GANs) [12] for the deblurring. Cho et al. [94] propose a multi-input multi-output U-net (MIMO-UNet) to improve the performance while reduce the computational cost. Zamir et al. [95] propose a multi-stage and multi-scale architecture for motion blur removal. And Chen et al. [96] propose a baseline model that achieves better performance and lower inference time compared with previous methods. But the above methods focus on the motion blur and therefore may be unsuitable for the defocus deblurring.

Recently, there are also some methods proposed for defoucs blurring in an end-to-end manner [33–35]. Abuolaim et al. [33] use the UNet structure to recover the sharp image in an end-to-end manner. Lee et al. [34] propose an iterative filter adaptive module to handle spatially-varying and large defocus blur and a training scheme based on defocus disparity estimation and re-blurring to boost the deblurring quality. Son et al. [35] propose to simulate inverse kernels by kernel sharing parallel atrous convolution block.

However, there still is a CNN-based method that follows first defocus map estimation and then non-blind deblurring [32]. Compared with end-to-end based methods, the two-step CNN-based methods can achieve higher performance since the bind deblurring problem is much more complicated than the non-blind deblurring problem, while the two steps CNN-based methods can utilize the important information from

the defocus map. After the estimation of the defocus map using convolutional neural network, Ma et al. [32] concatenates the defocus map and the defocus image as input of the neural network to achieve all sharp images. We argue that direct concatenation can not fully use the information in defocus map, which represents the blur level.

Considering the spatial variant property of the defocus blur and the blur level indicated in the defocus map, we employ the defocus map as the conditional guidance to adjust the features from the input blurring images instead of simple concatenation. Then we propose a simple but effective network with spatial modulation based on the defocus map. To achieve this, we design a network consisting of three sub-networks, including the defocus map estimation network, a condition network that encodes the defocus map into condition features and the defocus deblurring network that performs spatially dynamic modulation based on the condition features. And the spatially dynamic modulation is achieved by an affine transform function to adjust the features from the input blurry images. Experimental results show that our method can achieve better quantitative and qualitative evaluation performance than the existing state-of-the-art methods on the commonly used public test datasets.

4.2 Related Work

In this section, we briefly introduce the related works, including defocus map estimation, non-blind deblurring, and single image defocus deblurring.

4.2.1 Defocus Map Estimation

There are different approaches to defocus map estimation, which can be roughly divided into three categories: edge-based, region-based, and CNN-based methods.

The basic idea of the edge-based methods is to compute a sparse defocus map only at the edges of the images and then propagate the map to the whole image [28,97–100]. In [97], the input image is re-blurred by the Gaussian kernels, and then the defocus amount along the edges is computed by the rate of the gradients of the re-blurred image. Similar to [97], Elder and Zucker [98] propose a method that simultaneously detects image edges and estimates blur. These two studies [97, 98] only focus on the sparse defocus map estimation. Liu et al. [100] propose a two-parameter model

to improve the performance of the defocus map estimation on the edges. Bae and Durand [99] try to estimate the defocus map on the whole image. After the sparse defocus map estimation, their method uses a bilateral filter to remove outliers. It then uses a colorization-scheme-based interpolation method to achieve the full defocus map. Zhuo and Sim [28] propose using alpha Laplacian Matting to propagate the sparse defocus map to the whole image. These methods suffer from inaccurate defocus map estimation for the image areas far from the edges. In [29], a connected edge filter is proposed to smooth the initial sparse blur map based on pixel connectivity within detected edge contours. Then a fast-guided filter is used to propagate the sparse blur map through the whole image.

Region-based methods directly estimate the defocus amount from the local patches centered at the current pixel [101, 102]. For each local patch in the image, Trouvé et al. [101] use a maximum likelihood method to select the local blur from a set of PSF candidates. In [103], a machine learning approach based on the regression tree fields is used to train a model able to regress a coherent defocus blur map of the image, labeling each pixel by the scale of a defocus point spread function. Shi et al. [102] propose a method based on dictionary learning using sharp and slightly blurred patches.

Recently, CNN-based methods have been proposed for defocus map estimation. Yan and Shao [104] propose a method that first classifies the blur type and then estimates the blur parameter using a general regression neural network. Zhao et al. [105] propose a method that detects defocus blur by using a bottom-top-bottom fully convolutional network. Since defocus blur detection only classifies each pixel as blur pixel and the non-blur pixel while the defocus map estimation needs to estimate the blue level of each pixel so the defocus blur detection can be considered as the loose formulation of the defocus map estimation. Lee et al. [106] propose an end-to-end CNN-based method (DME-Net) for spatially varying defocus map estimation, for which they also created a synthetic dataset. They employ a domain adaptation method to address the gap between real and synthetic datasets. Theirs is the first truly CNN-based defocus map estimation method. Similar to [106], Ma et al., [32] train their model in an end-to-end manner for defocus map estimation. These methods suffer from the lack of sufficient real data.

4.2.2 Non-Blind Defocus Deblurring

Non-blind defocus deblurring, which assumes either the defocus map or the blur kernel to be known, is an ill-posed problem since some information will inevitably be lost during the blurring process. Most classical methods impose some image priors to regularize the solution, for example, patch-based prior [107], hyper-Laplacian prior [30], and local color prior [108]. These methods usually need computationally expensive iterative optimization. Recently, researchers have shown that CNN-based methods will be better than these classical approaches in terms of accuracy and efficiency [109–111]. However, most existing methods, such as Schuler et al. [109] and Xu et al. [110], need to be trained for each specific blur kernel; when we encounter images with unseen kernels, we need to retrain the networks.

4.2.3 Single Image Defocus Deblurring

Many studies have been conducted for single-image defocus deblurring. Conventional methods typically decompose the problem into two steps. The first step is to estimate the defocus map of an input image, which indicates the blur level for each pixel [27–29]. The second step is the non-blind deconvolution [30,31], where the defocus map estimated in the first step is used. Most of the methods employing this two-step approach focus on improving the accuracy of the first step, i.e., the defocus map estimation, since the small error on the defocus map will significantly deteriorate the final deblurring performance [29, 102, 103, 112].

Another approach is to train a network to directly predict the deblurred image from a blurry input image in an end-to-end fashion. Abuolaim et al. [33] use a network with the UNet structure to do this. Lee et al. [34] propose an iterative filter adaptive module to handle spatially-varying and large defocus blur. They also propose a training scheme based on defocus disparity estimation and re-blurring to boost deblurring quality. Son et al. [35] propose to simulate inverse kernels by kernel sharing parallel atrous convolution block. However, these methods based on direct prediction depend too much on the training data, and we believe their performance is suboptimal.

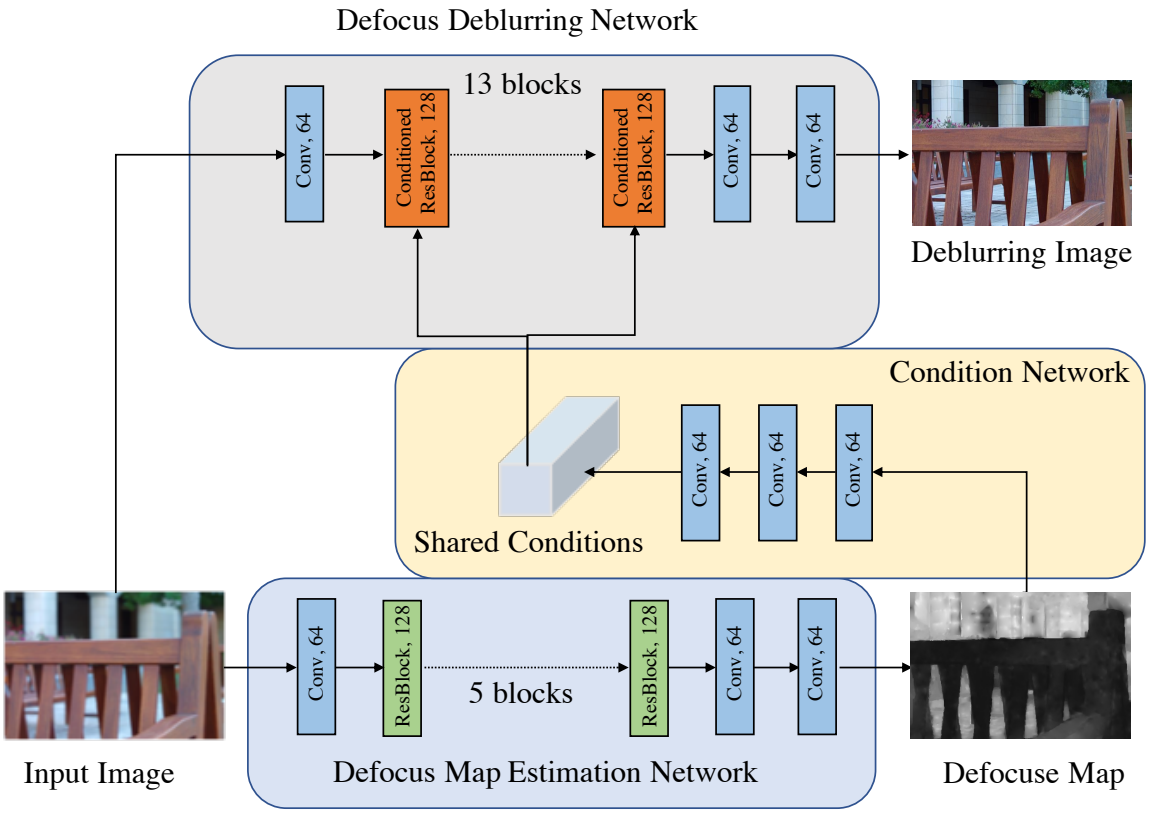


Figure 4.1: Architecture of the proposed network.

4.3 Proposed Method

We propose a network to remove spatial variant defocus blur from a single blurry image. Inspired by the previous deblurring methods that decompose the blind deblurring task into defocus map estimation task and non-blind deblurring [27–29, 32]. Our network also first estimates the defocus map as an intermediate result. For CNN-based based methods, they directly use the feed-forward networks to learn the mapping from the input blurring images, while the previous CNN-based two steps method directly concatenates the defocus map with the input blurring images as the input as the non-blind deblurring network. The defocus map, i.e., knowing the blur level for each pixel, is very important information for the non-blind deblurring. But previous method utilizes the defocus map by simple concatenation. We think this leads to suboptimal results.

Considering the spatial variant property of the defocus blur and the blur level indicated in the defocus map, we employ the defocus map as conditional guidance to

adjust the features from the input blurring images instead of simple concatenation. For better usage of the defocus map, we employ an affine transform to adjust the features from the input blurring images for each pixel. By considering the same blur level on regions in blurring image, we further decompose the parameters for scaling in the affine transform into spatial and channel dimensions to reduce the redundancy and improve the capacity.

Figure 4.1 shows our network which consists of three sub-network: defocus map estimation network, the condition network and the deblurring network.

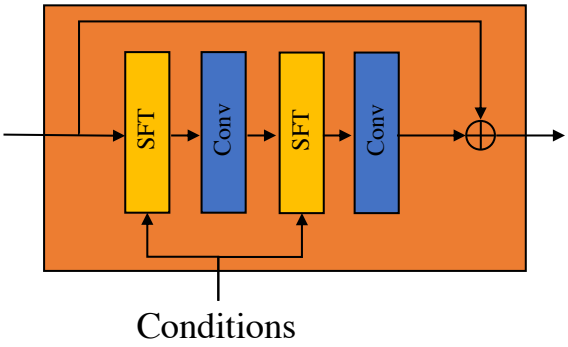


Figure 4.2: Architecture of the conditioned res-block.

4.3.1 Defocus Map Estimation Network

The defocus map estimation network utilizes several residual blocks to ease the training procedure and maximize the information flow, which takes the blurry images as input and then estimates the defocus map.

4.3.2 Condition Network

The condition network only consists of three convolutional layers, which takes the estimated defocus map as input and maps it into the feature space as conditions that are afterward used to modulate the intermediate features in the defocus deblurring network. The key to reconstructing the all-sharp images is to recover the missing details in the out-of-focus regions in the input blurrying images. Different areas in one image have different contents and blur levels. Further, different images also have different contents and blur levels. Therefore, it is necessary to deal with input images

with location-specific and image-specific operations. However, the convolutional layer is spatial invariant. Inspired by spatial feature transform (SFT) [113], we introduce a network with SFT to perform a spatial variant adjustment. Here the condition network is to generate the conditions for SFT from the defocus map.

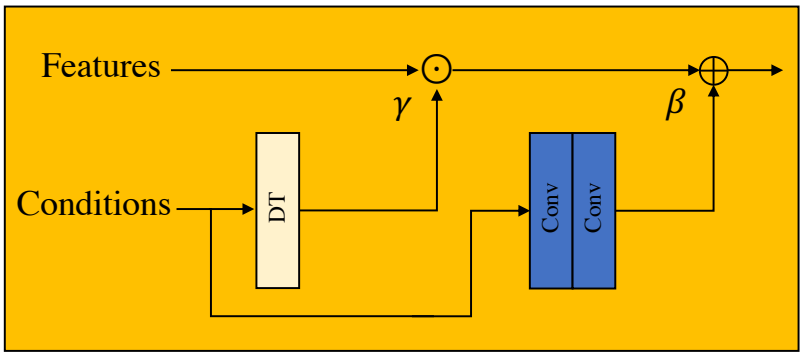


Figure 4.3: Architecture of the feature transform.

4.3.3 Defocus Deblurring Network

Defocus deblurring network takes not only the blurring image but also the conditions from the condition network as input. There are some conditioned residual blocks in the defocus deblurring network and the Figure 4.2 shows the details of conditioned residual blocks. The key component in conditioned residual blocks is the SFT layer. The structure of SFT layer is shown in Figure 4.3. The SFT learns a mapping function that generates the paired modulation parameters γ and β based on the defocus map as prior. The learned parameters adaptively adjust the output by an affine transform for each pixel to the intermediate features on the defocus deblurring network. Specifically, the SFT layer can be described as,

$$SFT(F) = \gamma \odot F + \beta \tag{4.2}$$

$F \in \mathbb{R}^{C\times H\times W}$ is the intermediate features in defocus deblurring network and $\gamma \in \mathbb{R}^{C\times H\times W}$ and $\beta \in \mathbb{R}^{C\times H\times W}$ are the parameters for modulation. $\odot$ is the element-wise multiplication.

Further, as shown in Figure 4.1, there are some regions with the same value in defocus map causing the redundancy on the conditions. Inspired by channel attention

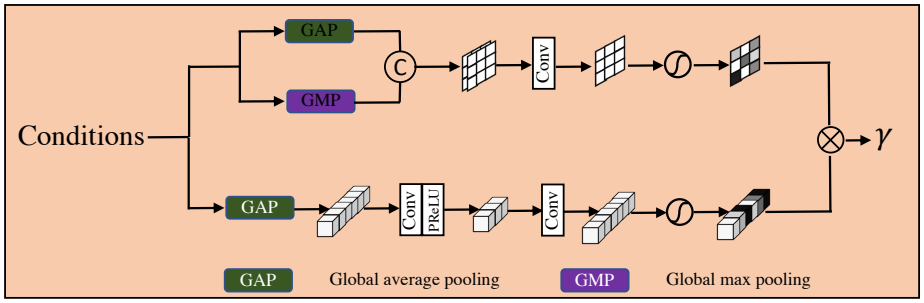


Figure 4.4: Architecture of the decomposition transform.

[68] and spatial attention [69], we perform decomposition on the generation for γ in the channel dimension $\mathbb{R}^{C\times1\times1}$ and spatial dimension $\mathbb{R}^{1\times H\times W}$ as shown in Figure 4.4. It should be noted that we do not apply the decomposition transform to the generation of β since there are differences for each pixel value, even though their blur levels are the same. Thus, the β without decomposition transform can provide the detailed compensation for the deblurring network. By using this modulation strategy, our method can better use the important information in defocus map.

4.3.4 Loss Function

The proposed network consists of three sub-network: defocus map estimation network, condition network and the deblurring network. Following the previous work [32], we use the L_1 norm loss and L_2 norm loss for training. Specifically, for defocus map estimation, the loss is computed as follows,

$$L_{dme} = ||DM_e - DM_{gt}||_1 \tag{4.3}$$

where DM_e is the estimated defocus map and the DM_{gt} is the ground truth for the defocus map.

While for the defocus deblurring, we exploit two losses: L_{db} and L_{wd}. L_{db} is defined as

$$L_{df} = ||I_{df} - I_{gt}||_1 \tag{4.4}$$

while L_{wd} is defined as

$$L_{wd} = ||W_{dm} * (I_{df} - I_{gt})||_2 \tag{4.5}$$

where I_{df} is the deblurring result of the deblurring network and the I_{gt} is the all-sharp ground truth image. W_{dm} is defined as

$$W_{dm} = \frac{DM_{gt}}{mean(DM_{gt})} \tag{4.6}$$

Similar to [32], we train our network for three stages. In stage one and stage two, the two networks are jointly trained for 400 and 200 epochs, respectively. For stage one, the ground truth defocus map is set as the input of the defocus deblurring network, which avoids divergence caused by random output of the defocus map estimation network. For stage two, the output of the defocus map estimation network is set as the input of the defocus deblurring network to jointly training the whole network. For both stage one and stage two, the loss used for training is

$$Loss_1 = \lambda_1 \times L_{dme} + \lambda_2 \times L_{df} \tag{4.7}$$

In stage three, we use the following loss to finetune the network for another 400 epochs,

$$Loss_2 = \lambda_2 \times L_{df} + \lambda_3 \times L_{wd} \tag{4.8}$$

In this work, we set the weights for loss function as $\lambda_1 = 0.2$, $\lambda_2 = 0.9$ and $\lambda_3 = 0.1$

4.4 Experiments

4.4.1 Experimental Settings

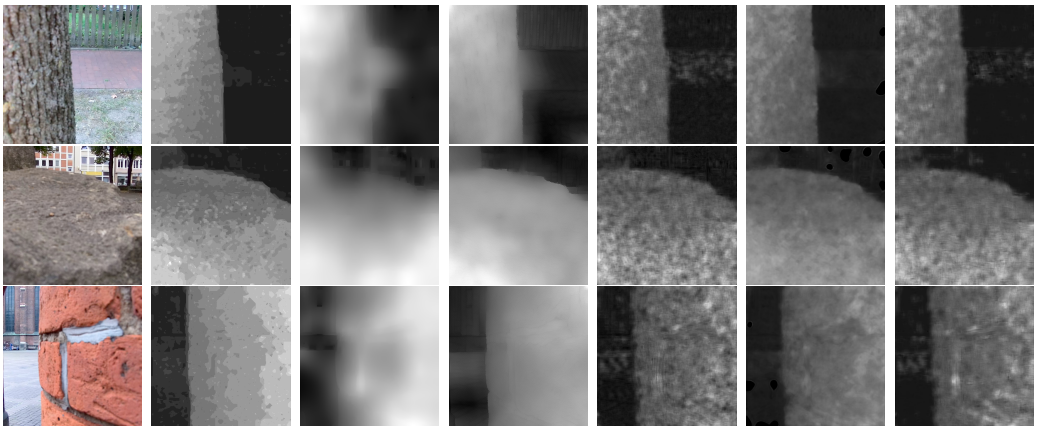
Training Data To train our network, we use the defocus image deblurring dataset in [32] which consists of both the defocus map ground truth and all-sharp image ground truth.

Test Data Followed the previous work [32], we evaluate the proposed method on the Realistic dataset [103] and the DED test dataset [32].

Implementation Details For training, we use the Adam optimizer [45] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ with initial learning rate 1×10^{-4} and set the batch size to 16. And the number of epochs for each stage is mentioned above. The input blurring

Table 4.1: Quantitative Results for Defocus Map Estimation

Methods	MAE, Realistic	MSE, Realistic
Karraali [29]	0.3102	0.1245
DMENet [106]	**0.1191**	**0.0242**
DID-ANet [32]	0.1331	0.0312
Ours-DMENet	0.1242	0.0281
Ours	0.1303	0.0299



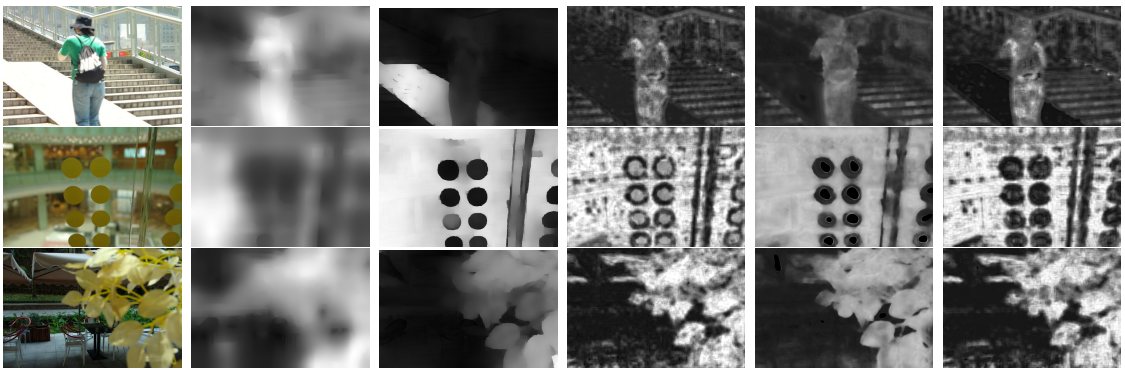
(a) Input image (b) Ground truth (c) Karraali [29] (d) DMENet [106] (e) DID-ANet [32] (f) Ours-DMENet (g) Ours

Figure 4.5: Visual comparison of defocus map estimation on realistic.

image, the defocus map ground truth and the corresponding all-sharp ground truth are randomly cropped into patches with 256×256. We also apply other data augmentation like random rotation and flipping to avoid over-fitting. We implement our model by PyTorch [72] platform and train our model on NVIDIA GeForce RTX 2080 GPUs.

4.4.2 Experimental Resutls

We first evaluate the performance of defocus map estimation. We compare the proposed method with the methods of Karaali and Jung [29] and the recent deep learning-based DME-Net [106] and DID-ANet [32] for defoucs map estimation. Mean absolute error (MAE) and mean squared error (MSE) are used as the evaluation metrics. Table 4.1 shows the quantitative results on Realistic dataset. The proposed



(a) Input image (b) Ground truth (c) Karraali [29] (d) DMENet [106] (e) DID-ANet [32] (f) Ours

Figure 4.6: Visual comparison of defocus map estimation on DED dataset.

method achieves almost the same performance with DID-ANet [32] since we adopt the same network with DID-ANet [32] and is comparable with DMENet [106] on Realistic dataset.

Several samples for visualization of defocus map estimation are shown in Figure 4.5 and 4.6.

We compare the proposed method with the DME-Net [106] that estimates the defocus map by CNN and achieves deblurring by conventional deconvolution [30], two CNN-based methods for defocus deblurring (DPDDNet [33] and IFAN [34]), three CNN-based state-of-arts methods (MPRNet [95], MIMO-UNet [94] and NAFT [96]) for motion deblurring and two-step CNN-based method (DID-ANet [32]). All the methods are fine-tuned on the training set of DED dataset. And the training settings are set separately based on the original paper. The epochs for fine-tuning are set to 600 to ensure the models are convergent.

We use the Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) for evaluation metrics. The quantitative results are shown in Table 4.2 where the best results are in bold. For Realistic dataset, the proposed method achieves better performance than the other methods in terms of PSNR and SSIM. The CNN-based methods for motion blur and defocus blur do not achieve good performance compared with the two-stage methods (e.g., DID-ANet [32]), since they do not utilize the information from the defocus map which is essential for defocus image blurring. Although The proposed method achieves almost the same performance with DID-ANet [32] in the defoucs map estimation, it still achieves much higher performance

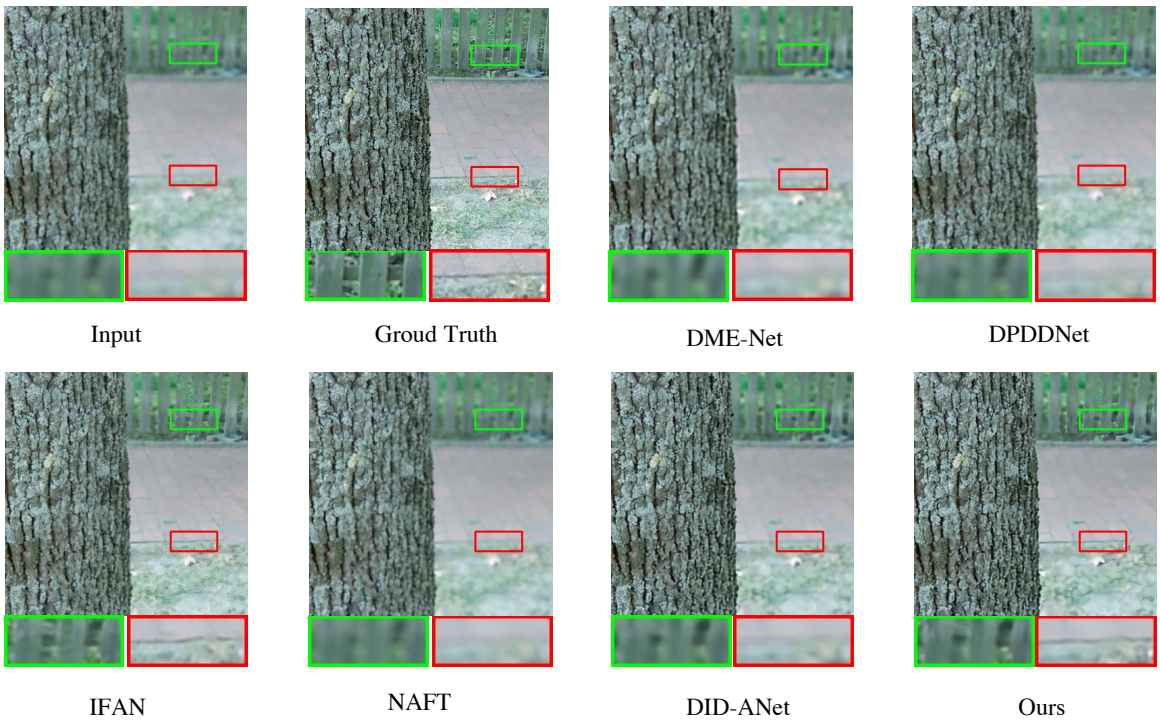


Figure 4.7: Visual comparison of defocus image deblurring on realistic (08).

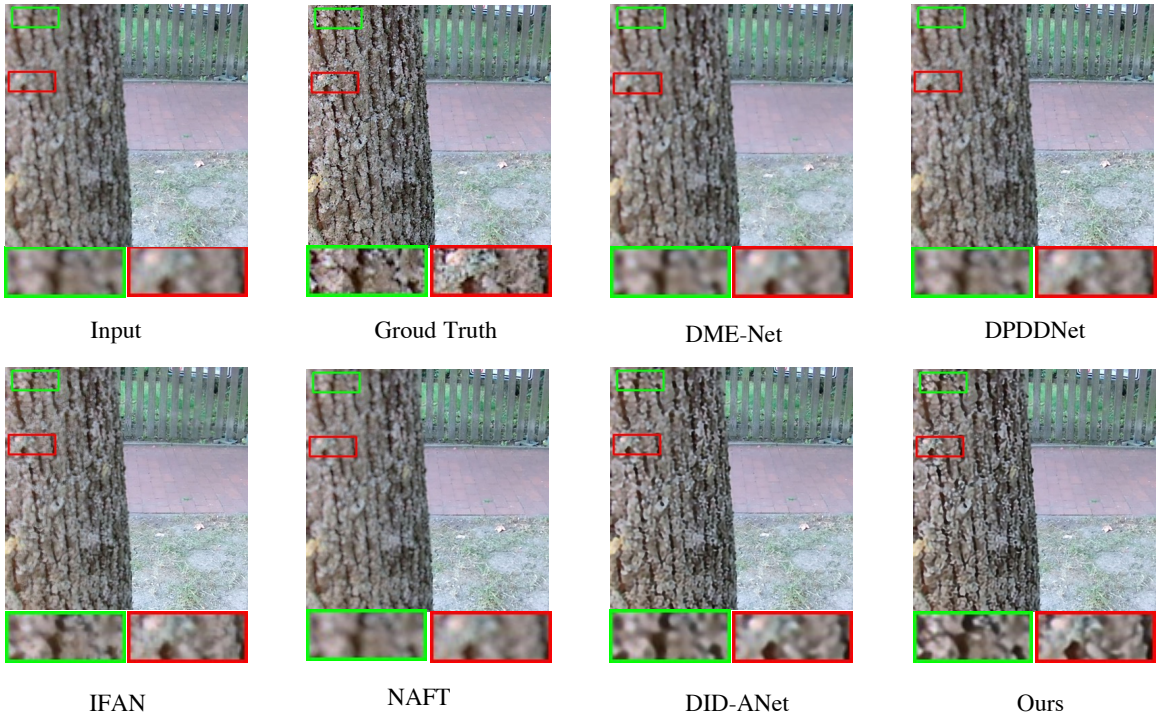


Figure 4.8: Visual comparison of defocus image deblurring on realistic (09).

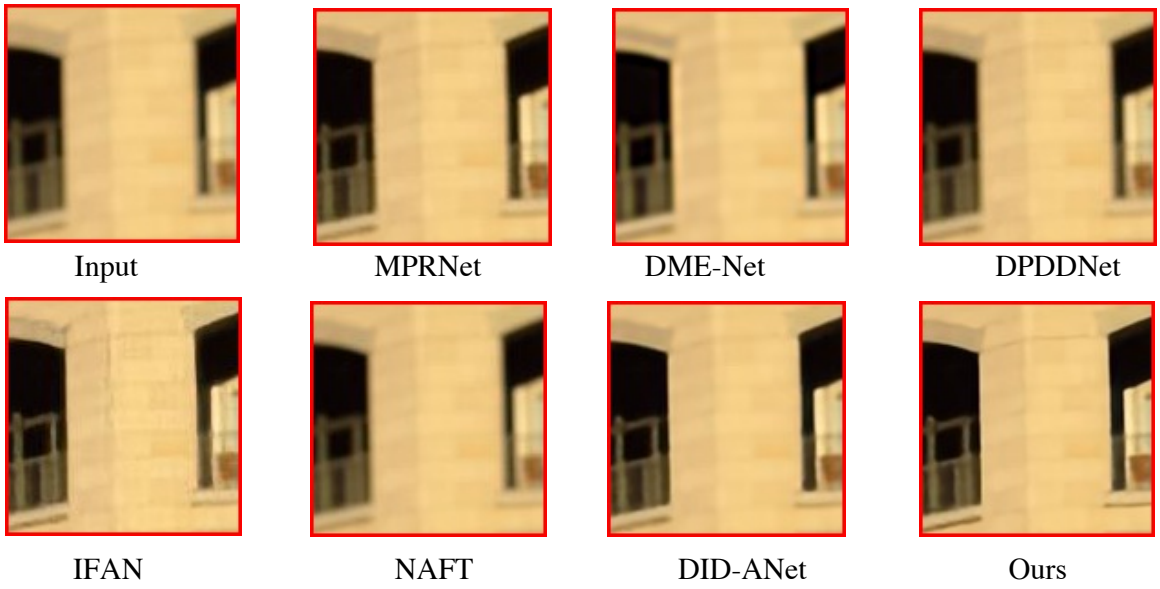


Figure 4.9: Visual comparison of defocus image deblurring on DED dataset (604).

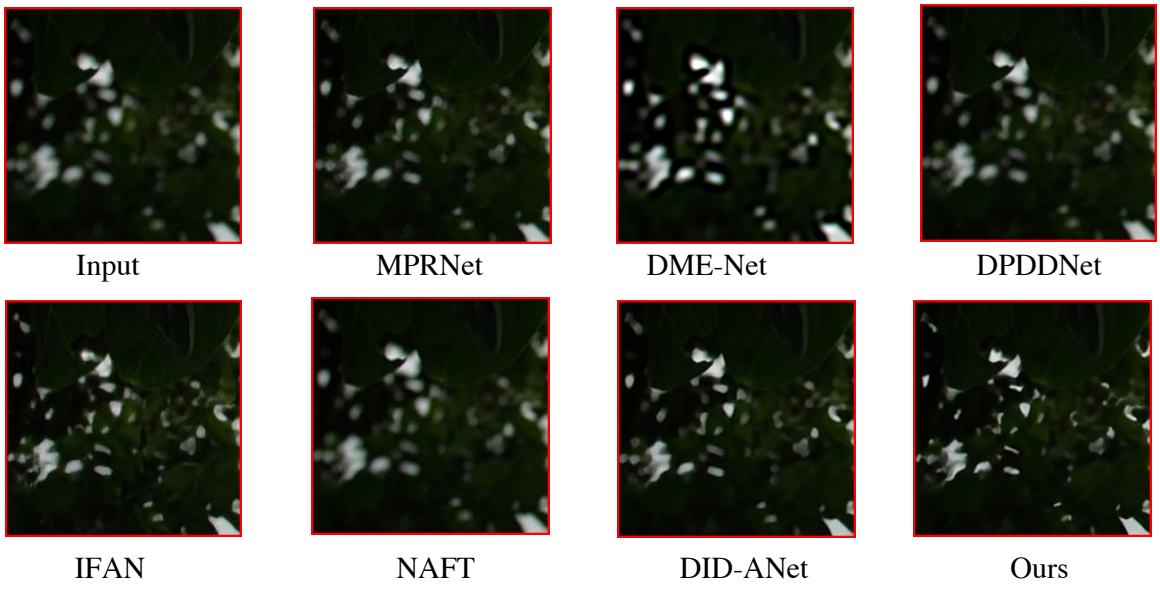


Figure 4.10: Visual comparison of defocus image deblurring on DED dataset (794).

Figure 4.11: Visual comparison of defocus image deblurring on DED dataset (971).

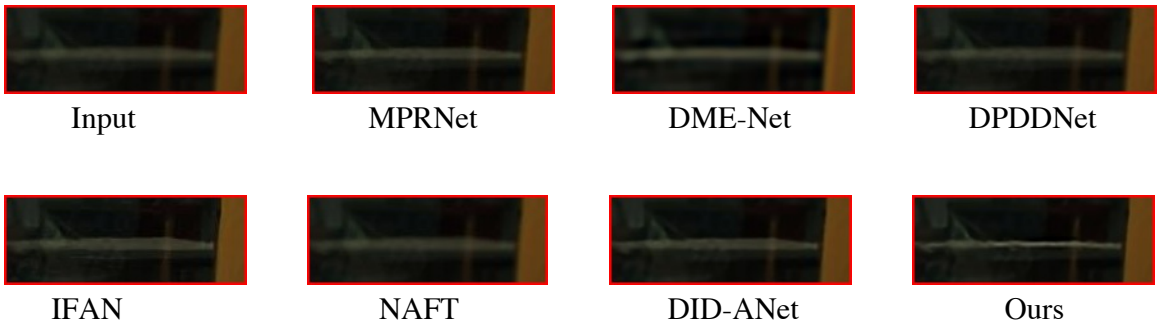


Figure 4.12: Visual comparison of defocus image deblurring on DED dataset (941).

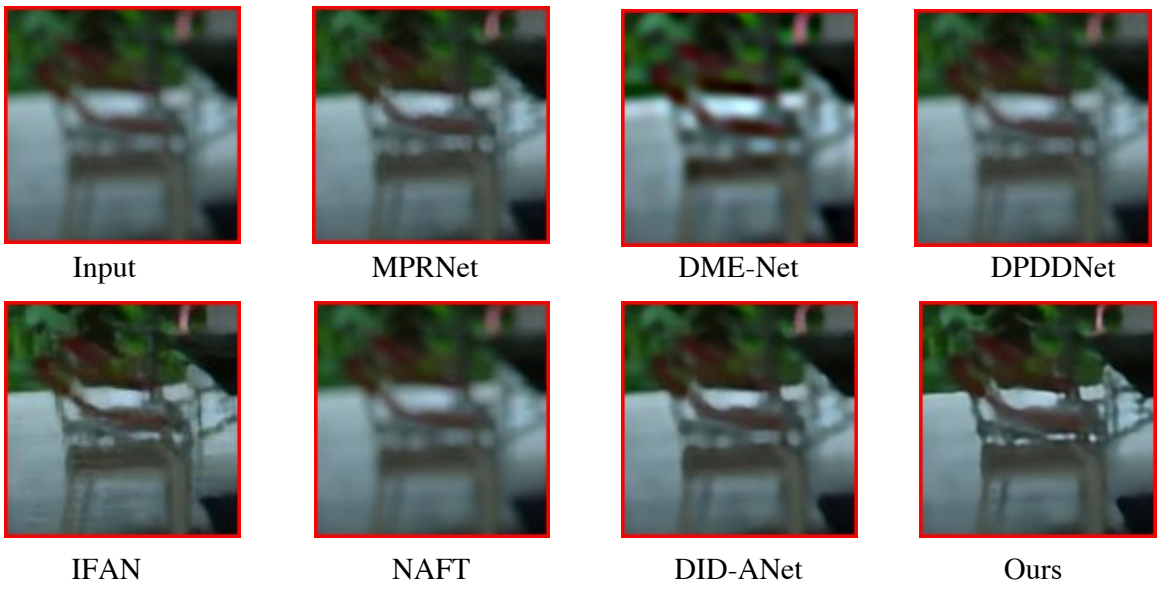


Figure 4.13: Visual comparison of defocus image deblurring on DED dataset (1007).

than DID-ANet [32] with more than 0.6db improvement, which shows the effectiveness of the proposed method. And also compared DME-Net [106] with conventional deconvolution [30], the CNN-based methods (e.g., DID-ANet [32]) can achieve much better performance.

Several visualization results are shown in Figure 4.7-4.11. Among them, Figure 4.7 and 4.8 are from Realistic test set while Figure 4.9-4.11 are from DED test set. Since the image size for the DED test set is large, we crop the images from DED test set and zoom them for a better view.

The images from Figure 4.7 and 4.8 are almost the same contents but have different focal plane. The image (08) in the Figure 4.7 focuses on the tree then the background is blurry. As shown in Figure 4.7, the structures on the road and the fence are clearer in our result. While in Figure 4.8, ours results have clearer texture compared with other methods. As shown in Figure 4.9, our method can achieve the sharper edge on the wall with fewer artifacts. The light region in input image shown in Figure 4.10 is blurry, while our method can successfully recover the sharp shape of the light region. The hand and the watch are more realistic and clearer in our result.

In contrast, the previous works cannot address the defocus blur very well. The DME-Net [106] estimates the defocus map and then uses the deconvolution deblurring

Table 4.2: Quantitative Results for Defocus Image Deblurring

	PSNR, Realistic	SSIM, Realistic
DMENet [106]	24.0397	0.7180
DPDDNet [33]	24.7232	0.7640
IFAN [34]	24.9261	0.8211
MPRNet [95]	25.2738	0.7869
MIMO-UNet [94]	25.3430	0.7962
NAFT [96]	24.3461	0.7484
DID-ANet [32]	25.9491	0.8204
Ours	**26.5535**	**0.8465**

[30] to achieve defocus deblurring. Their results look blurry compared with ours. On the other hand, the CNN-based based methods (DPDDNet [33] and IFAN [34]) also achieve blurry results though the model are finetuned on the DED dataset. For the state-of-the-art method in motion deblurring methods, NAFT [96] cannot deal with the defocus blur very well. For the two-stage methods, DID-ANet achieves better performance than the above methods. However, the important information in the defocus map can not be fully utilized by simple concatenation. On the contrary, the proposed method exploits the information from the defocus map by the SFT with the decomposition method, which leads to higher PSNR and SSIM than previous methods, as shown in Table 4.2

4.4.3 Ablation Study

We demonstrate the effectiveness of each component in the proposed method on the Realistic dataset. The results are shown in Table 4.3. For the baseline model, we remove the SFT with the decomposition method. For the "baseline+SFT", it means that we use the original SFT in the baseline. For the "baseline+SFT-Dec", it means that we use the SFT with the decomposition on γ into the baseline. For the "baseline+SFT-FDec", it means that we use the SFT with the decomposition on both γ and β into the baseline. S1, S2 and S3 indicate the different training stages,

Table 4.3: Ablation Study on Results for Defocus Image Deblurring

Methods	PSNR, Realistic	SSIM, Realistic
baseline-S1	25.6443	0.8077
+SFT-S1	26.1286	0.8294
+SFT-Dec-S1	26.0837	0.8289
+SFT-FDec-S1	25.9939	0.8253
+SFT-Dec-DME-S1	26.4595	0.8402
baseline-S2	25.7936	0.8118
+SFT-S2	26.2533	0.8307
+SFT-Dec-S2	26.4275	0.8310
+SFT-FDec-S2	26.1735	0.8307
+SFT-Dec-DME-S2	**26.6031**	**0.8482**
baseline-S3	25.9638	0.8191
+SFT-S3	26.3973	0.8379
+SFT-Dec-S3	**26.5535**	**0.8465**
+SFT-FDec-S3	26.4327	0.8400
+SFT-Dec-DME-S3	**26.61**	0.8455
+SFT-Dec-end	26.1143	0.8259

respectively while "-end" means the model is trained in an end-to-end manner with the supervision of the defocus map and the all sharp ground truth.

As shown in Table 4.3, for all these methods, the performance increases after each stage training. While the final model "baseline+SFT-Dec-s3" achieves the best performance compared with other ablated networks. At the same time, if the final model is trained in an end-to-end manner, the performance decreases by a large margin. By introducing the SFT, the performance increases about 0.4db in PSNR and 0.02 in SSIM compared with the baseline model. While incorporating the decomposition method, the performance increases by about 0.6db in PSNR and 0.03 in SSIM compared with the baseline model. However, if we also apply the decomposition method to the β, the performance drops about 0.1db in PSNR. This is because the β provides

the detailed information for modulation but the decomposition method will remove these details to some extent.

We also use the DMENet to estimate the defocus map as indicated "+SFT-Dec-DME". As shown in Table 4.3, the "+SFT-Dec-DME" can achieve the better performance than the "+SFT-Dec". However, the number of parameter in "+SFT-Dec-DME" is 30.20M while the number of parameter in "+SFT-Dec" is 5.12M. And the improvement by using DMENet for defocus map estimation is only 0.06db. So considering the trade-off between the performance and the model complexity, we use the +SFT-Dec" as our full model.

4.5 Summary and Conclusion

In this section, we propose a new method for defocus image deblurring by first defocus map estimation and then the defocus deblurring direction. The defocus map is important information for defocus image deblurring since it contains the blur level for each pixel. To remove the spatial variant blur, we introduce the spatial feature transform and the decomposition technique to perform spatial modulation based on the defocus map. The experimental results have validated the effectiveness of the proposed approach.

Chapter 5

Conclusion

In this dissertation, we have studied image quality enhancement from limited devices. Specifically, we utilize the CNN-based methods for high dynamic range imaging and defocus image deblurring. We have proposed a novel method that can better fuse the features based on two ideas for high dynamic range imaging. One is multi-step feature fusion; our network gradually fuses the features in a stack of blocks having the same structure. The other is the design of the component block that effectively performs two operations essential to the problem, i.e., comparing and selecting appropriate images/regions. Experimental results show that the proposed method outperforms the previous state-of-the-art methods on the standard benchmark tests. This is introduced in Chapter 2.

In Chapter 3, we have further proposed a network that follows an alignment-before-merging manner. Specifically, we propose a deep network that tries to learn multi-scale feature flow guided by the regularized loss. It first extracts multi-scale features and then aligns features from non-reference images. After alignment, we use residual channel attention blocks to merge the features from different images. Extensive qualitative and quantitative comparisons show that our approach achieves state-of-the-art performance and produces excellent results where color artifacts and geometric distortions are significantly reduced.

In Chapter 4, we have proposed a network for defocus image deblurring. Previous classical methods follow two-step approaches, i.e., first defocus map estimation and then the non-blind deblurring. In the era of deep learning, some researchers try to

address these two problems by CNN. However, the simple concatenation of defocus map, which represents the blur level, leads to suboptimal performance. Considering the spatial variant property of the defocus blur and the blur level indicated in the defocus map, we employ the defocus map as conditional guidance to adjust the features from the input blurring images instead of simple concatenation. Then we propose a simple but effective network with spatial modulation based on the defocus map. To achieve this, we design a network consisting of three sub-networks, including the defocus map estimation network, a condition network that encodes the defocus map into condition features and the defocus deblurring network that performs spatially dynamic modulation based on the condition features. And the spatially dynamic modulation is based on an affine transform function to adjust the features from the input blurry images. Experimental results show that our method can achieve better quantitative and qualitative evaluation performance than the existing state-of-the-art methods on the commonly used public test datasets.

Appendix A

Appendix for Multi-Scale Feature Flow Network (Chapter 3)

A.1 Visualization Results

We provide additional examples of the results obtained by our method and existing state-of-the-art methods in Figs. A.1. Specifically, we compare our model with two single image HDR imaging methods (i.e., TMO [61] and HDRCNN [60]) and five multi-image HDR imaging methods (i.e., the patch-based method [2], the flow-based method with CNN merger [1], the U-net structure without optical flow [5], the attention-guide method (AHDR) [54], and progressive and selective fusion network (PSFNet) [73]).

As shown in Figure A.2 and A.3, there are two samples to illustrate the effectiveness of the regularization loss. Here we only show two settings for λ, i.e., $\lambda = 0$ and $\lambda = 2$. As show in Figure A.2 and A.3, sub-figure (a) shows the low exposure image, sub-figure (b) shows the medium exposure image, sub-figure (c) shows the low exposure image warped by the flow learned in $\lambda = 0$ settings and sub-figure (d) shows the low exposure image warped by the flow learned in $\lambda = 2$ settings.

Figure A.1: Results from the Testing data (BarbequeDay) of [1]. Upper row from left to right: the two input LDR images, the HDR image produced by the proposed method, and (zoomed-in) LDR image patches with two identical positions/sizes (in green and red). Lower row: the same patches of the HDR images produced by different existing methods

(a) EV = -2.0

(b) EV = 0

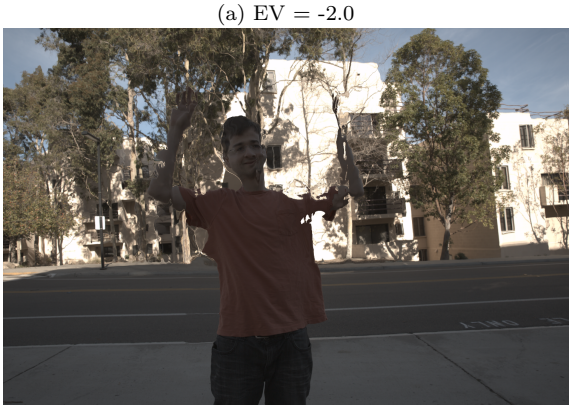
(c) $\lambda = 0$

(d) $\lambda = 2$

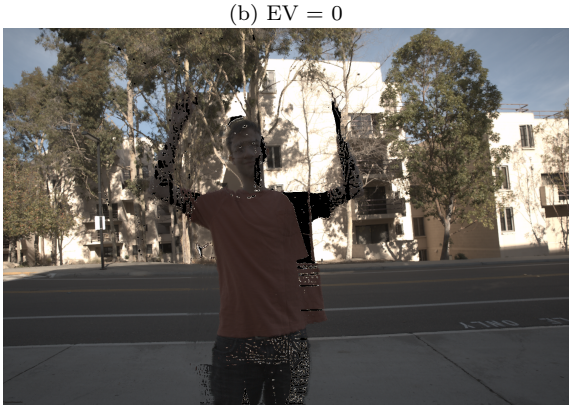
Figure A.2: The low exposure images (010) in test set of [1] warped by the flow for different λ values.

(a) EV = -2.0

(b) EV = 0

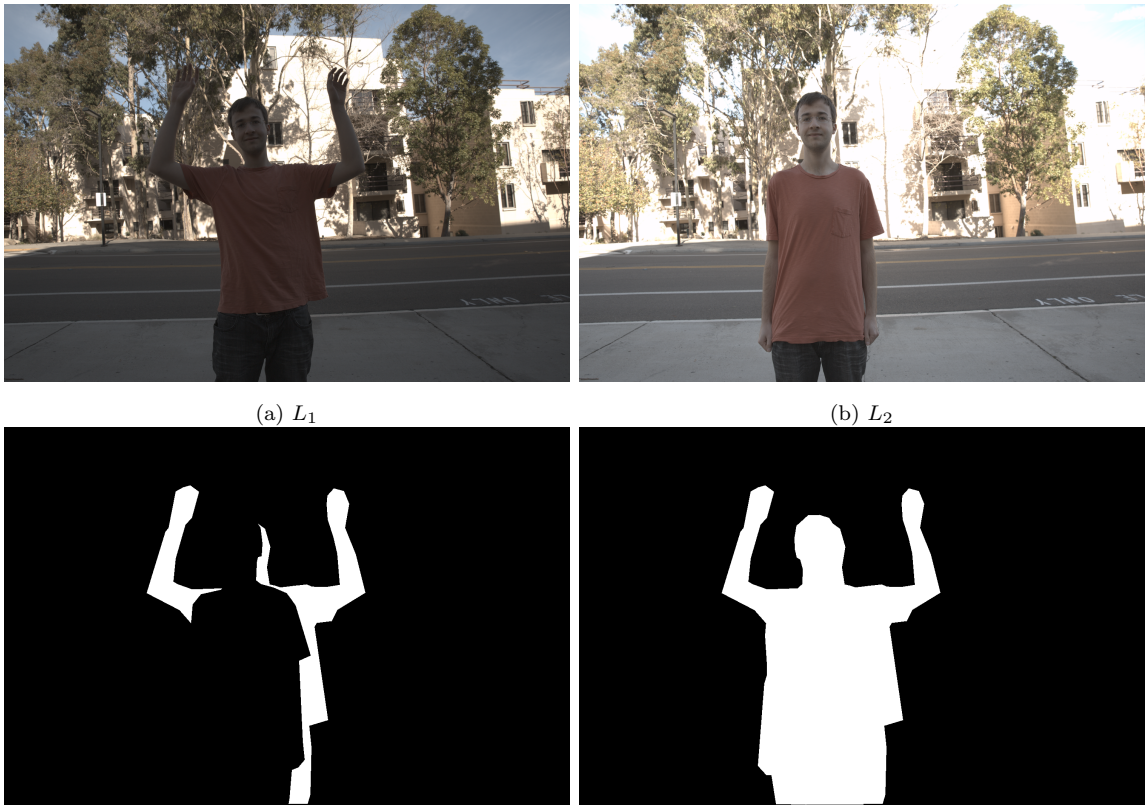


(c) $\lambda = 0$



(d) $\lambda = 2$

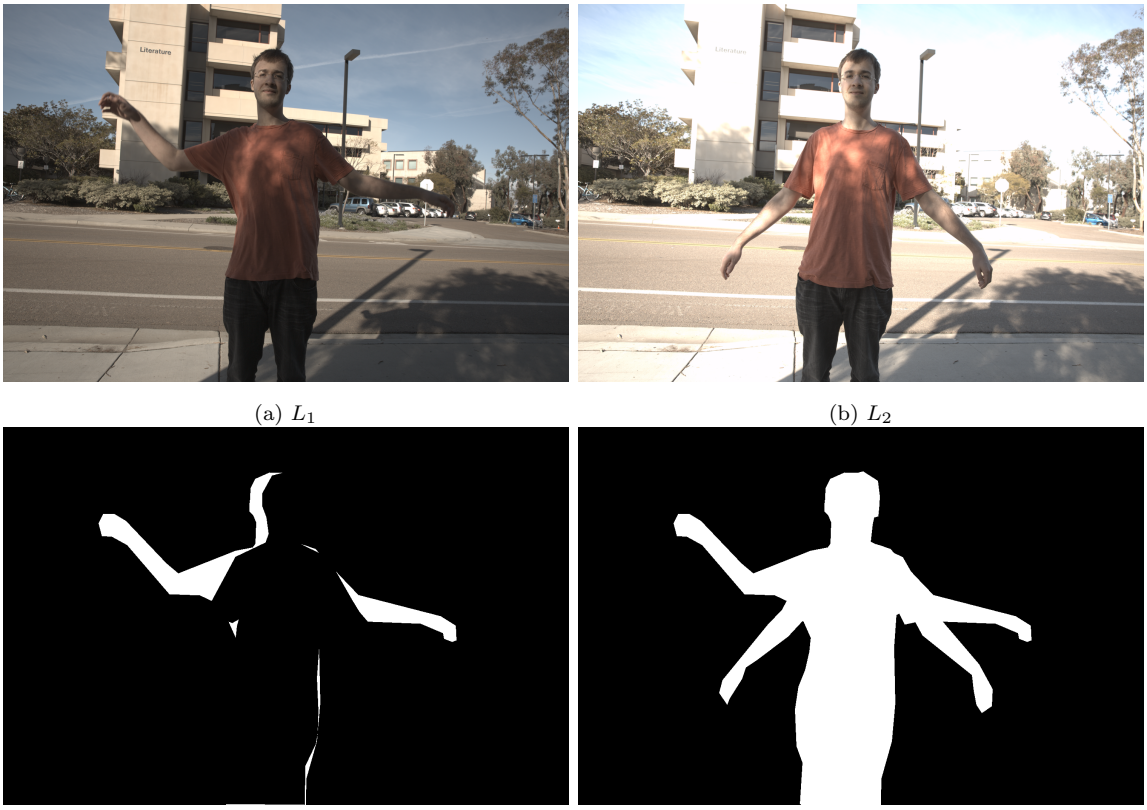
Figure A.3: The low exposure images (007) in test set of [1] warped by the flow for different λ values.



(a) L_1 (b) L_2

(c) Mask1 (d) Mask2

Figure A.4: Two different masks on Testing data (07) in [1].



(a) L_1 (b) L_2

(c) Mask1 (d) Mask2

Figure A.5: Two different masks on Testing data (09) in [1].

(a) L_1 (b) L_2

(c) Mask1 (d) Mask2

Figure A.6: Two different masks on Testing data (10) in [1].

Bibliography

[1] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep high dynamic range imaging of dynamic scenes. *ACM Transactions on Graphics (TOG)*, 36(4):144–1, 2017.

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

[3] Okan Tarhan Tursun, Ahmet Oğuz Akyüz, Aykut Erdem, and Erkut Erdem. An objective deghosting quality metric for hdr images. In *Computer Graphics Forum*, volume 35, pages 139–152. Wiley Online Library, 2016.

[4] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[5] Shangzhe Wu, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Deep high dynamic range imaging with large foreground motions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 117–132, 2018.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2017.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[13] Jiang Duan, Marco Bressan, Chris Dance, and Guoping Qiu. Tone-mapping high dynamic range images by novel histogram adjustment. *Pattern Recognition*, 43(5):1847–1862, 2010.

[14] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced High Dynamic Range Imaging, Second Edition*. A. K. Peters, Ltd., USA, 2nd edition, 2017.

[15] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 267–276, 2002.

[16] Yeyao Chen, Gangyi Jiang, Mei Yu, You Yang, and Yo-Sung Ho. Learning stereo high dynamic range imaging from a pair of cameras with different exposure parameters. *IEEE Transactions on Computational Imaging*, 6:1044–1058, 2020.

[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. Wiley Online Library, 2009.

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

[19] Abhishek Badki, Nima Khademi Kalantari, and Pradeep Sen. Robust radiometric calibration for dynamic scenes in the wild. In *2015 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2015.

[20] 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 Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

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

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

[23] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[24] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[25] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[26] Michael Potmesil and Indranil Chakravarty. Synthetic image generation with a lens and aperture camera model. *ACM Transactions on Graphics (TOG)*, 1(2):85–108, 1982.

[27] Yu-Wing Tai and Michael S Brown. Single image defocus map estimation using local contrast prior. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 1797–1800. IEEE, 2009.

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

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

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

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

[32] Haoyu Ma, Shaojun Liu, Qingmin Liao, Juncheng Zhang, and Jing-Hao Xue. Defocus image deblurring network with defocus map estimation as auxiliary task. *IEEE Transactions on Image Processing*, 31:216–226, 2021.

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

[34] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2034–2042, 2021.

[35] Hyeongseok Son, Junyong Lee, Sunghyun Cho, and Seungyong Lee. Single image defocus deblurring using kernel-sharing parallel atrous convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2642–2650, 2021.

[36] Alan C Brooks, Xiaonan Zhao, and Thrasyvoulos N Pappas. Structural similarity quality metrics in a coding context: exploring the space of realistic distortions. *IEEE Transactions on image processing*, 17(8):1261–1273, 2008.

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, 2015.

[39] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International conference on machine learning*. PMLR, 2010.

[40] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[42] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[43] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. A comprehensive evaluation of full reference image quality assessment algorithms. In *2012 19th IEEE International Conference on Image Processing*, pages 1477–1480. IEEE, 2012.

[44] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[46] Allan G Rempel, Wolfgang Heidrich, Hiroe Li, and Rafał Mantiuk. Video viewing preferences for hdr displays under varying ambient illumination. In *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization*, pages 45–52, 2009.

[47] Eike Falk Anderson and Leigh McLoughlin. Critters in the classroom: a 3d computer-game-like tool for teaching programming to computer animation students. In *ACM SIGGRAPH 2007 Educators Program*, page 7. 2007.

[48] Erum Arif Khan, Ahmet Oguz Akyuz, and Erik Reinhard. Ghost removal in high dynamic range images. In *2006 International Conference on Image Processing*, pages 2005–2008. IEEE, 2006.

[49] Yong Seok Heo, Kyoung Mu Lee, Sang Uk Lee, Youngsu Moon, and Joonhyuk Cha. Ghost-free high dynamic range imaging. In *Asian Conference on Computer Vision*, pages 486–500. Springer, 2010.

[50] Qingsen Yan, Jinqiu Sun, Haisen Li, Yu Zhu, and Yanning Zhang. High dynamic range imaging by sparse representation. *Neurocomputing*, 269:160–169, 2017.

[51] Luca Bogoni. Extending dynamic range of monochrome and color images through fusion. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 7–12. IEEE, 2000.

[52] Jun Hu, Orazio Gallo, Kari Pulli, and Xiaobai Sun. Hdr deghosting: How to deal with saturation? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1163–1170, 2013.

[53] David Hafner, Oliver Demetz, and Joachim Weickert. Simultaneous hdr and optic flow computation. In *2014 22nd International Conference on Pattern Recognition*, pages 2065–2070. IEEE, 2014.

[54] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1751–1760, 2019.

[55] Qingsen Yan, Lei Zhang, Yu Liu, Yu Zhu, Jinqiu Sun, Qinfeng Shi, and Yanning Zhang. Deep hdr imaging via a non-local network. *IEEE Transactions on Image Processing*, 29:4308–4322, 2020.

[56] Katrien Jacobs, Celine Loscos, and Greg Ward. Automatic high-dynamic range image generation for dynamic scenes. *IEEE Computer Graphics and Applications*, 28(2):84–93, 2008.

[57] Wei Zhang and Wai-Kuen Cham. Gradient-directed multiexposure composition. *IEEE Transactions on Image Processing*, 21(4):2318–2323, 2011.

[58] Chul Lee, Yuelong Li, and Vishal Monga. Ghost-free high dynamic range imaging via rank minimization. *IEEE Signal Processing Letters*, 21(9):1045–1049, 2014.

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

[60] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Transactions on Graphics (TOG)*, 36(6):1–15, 2017.

[61] Yuki Endo, Yoshihiro Kanamori, and Jun Mitani. Deep reverse tone mapping. *ACM Transactions on Graphics (Proc. of SIGGRAPH ASIA 2017)*, 36(6), November 2017.

[62] Zhiyuan Pu, Peiyao Guo, M Salman Asif, and Zhan Ma. Robust high dynamic range (hdr) imaging with complex motion and parallax. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[63] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.

[64] Charles Poynton. *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.

[65] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European conference on computer vision (ECCV)*, pages 402–419. Springer, 2020.

[66] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 510–519, 2019.

[67] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020.

[68] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[69] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[70] Rafał Mantiuk, Kil Joong Kim, Allan G Rempel, and Wolfgang Heidrich. Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics (TOG)*, 30(4):1–14, 2011.

[71] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[72] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[73] Qian Ye, Jun Xiao, Kin-man Lam, and Takayuki Okatani. Progressive and selective fusion network for high dynamic range imaging. In *Proceedings of the ACM International Conference on Multimedia*, 2021.

[74] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4947–4956, 2021.

[75] Jiayi Lin, Yan Huang, and Liang Wang. Fdan: Flow-guided deformable alignment network for video super-resolution. *arXiv preprint arXiv:2105.05640*, 2021.

[76] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[77] Sungil Choi, Jaehoon Cho, Wonil Song, Jihwan Choe, Jisung Yoo, and Kwanghoon Sohn. Pyramid inter-attention for high dynamic range imaging. *Sensors*, 20(18):5102, 2020.

[78] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[79] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep hdr video from sequences with alternating exposures. In *Computer graphics forum*, volume 38, pages 193–205. Wiley Online Library, 2019.

[80] Jason J Yu, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[81] K Ram Prabhakar, Rajat Arora, Adhitya Swaminathan, Kunal Pratap Singh, and R Venkatesh Babu. A fast, scalable, and reliable deghosting method for extreme exposure fusion. In *2019 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2019.

[82] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017.

[83] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

[84] K Ram Prabhakar, Susmit Agrawal, Durgesh Kumar Singh, Balraj Ashwath, and R Venkatesh Babu. Towards practical and efficient high-resolution hdr deghosting with cnn. In *European Conference on Computer Vision*, pages 497–513. Springer, 2020.

[85] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2016.

[86] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012.

[87] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[88] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014.

[89] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[90] Patrizio Campisi and Karen Egiazarian. *Blind image deconvolution: theory and applications*. CRC press, 2017.

[91] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[92] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.

[93] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019.

[94] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4641–4650, 2021.

[95] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021.

[96] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022.

[97] Alex Paul Pentland. A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, (4):523–531, 1987.

[98] James H Elder and Steven W Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on pattern analysis and machine intelligence*, 20(7):699–716, 1998.

[99] Soonmin Bae and Frédo Durand. Defocus magnification. In *Computer graphics forum*, volume 26, pages 571–579. Wiley Online Library, 2007.

[100] Shaojun Liu, Fei Zhou, and Qingmin Liao. Defocus map estimation from a single image based on two-parameter defocus model. *IEEE Transactions on Image Processing*, 25(12):5943–5956, 2016.

[101] Pauline Trouvé, Frédéric Champagnat, Guy Le Besnerais, and Jérôme Idier. Single image local blur identification. In *2011 18th IEEE International Conference on Image Processing*, pages 613–616. IEEE, 2011.

[102] Jianping Shi, Li Xu, and Jiaya Jia. Just noticeable defocus blur detection and estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 657–665, 2015.

[103] Laurent D'Andrès, Jordi Salvador, Axel Kochale, and Sabine Süsstrunk. Non-parametric blur map regression for depth of field extension. *IEEE Transactions on Image Processing*, 25(4):1660–1673, 2016.

[104] Ruomei Yan and Ling Shao. Blind image blur estimation via deep learning. *IEEE Transactions on Image Processing*, 25(4):1910–1921, 2016.

[105] Wenda Zhao, Fan Zhao, Dong Wang, and Huchuan Lu. Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3080–3088, 2018.

[106] Junyong Lee, Sungkil Lee, Sunghyun Cho, and Seungyong Lee. Deep defocus map estimation using domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12222–12230, 2019.

[107] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486. IEEE, 2011.

[108] Neel Joshi, C Lawrence Zitnick, Richard Szeliski, and David J Kriegman. Image deblurring and denoising using color priors. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1550–1557. IEEE, 2009.

[109] Christian J Schuler, Harold Christopher Burger, Stefan Harmeling, and Bernhard Scholkopf. A machine learning approach for non-blind image deconvolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1067–1074, 2013.

[110] Li Xu, Jimmy S Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. *Advances in neural information processing systems*, 27, 2014.

[111] Jiawei Zhang, Jinshan Pan, Wei-Sheng Lai, Rynson WH Lau, and Ming-Hsuan Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3817–3825, 2017.

[112] Jinsun Park, Yu-Wing Tai, Donghyeon Cho, and In So Kweon. A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1736–1745, 2017.

[113] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

Acknowledgments

Firstly, I would like to express my sincere appreciation to my advisor Professor Takayuki Okatani for his priceless suggestions and comments on my research. Professor Okatani has been very supportive during the last three years and gives me the freedom to choose the topic that I am interested in. And also, I have learned much invaluable knowledge from Professor Okatani, such as writing skills, ways of scientific thinking, and so on. I also thank Assistant Professor Suganuma Masanori and Research Assistant Professor Liu Xing for maintaining the computing system which plays an important role in my research and for their suggestions and comments on my research.

And I would also like to thank Zhijie Wang, Kang-Jun Liu, Wenzheng Song, Jie Zhang, Xiangyong Lu, and other members of our laboratory for their help during my study. I would also thank my friend Jun Xiao for his help in my study.

I would thank lina zeng who is a lovely girl and helps me a lot on my difficult days.

I also want to thank my family for their unconditional support and love.

Thank you all!