IMPROVING IMAGE QUALITY BY DYNAMIC RANGE AND RESOLUTION ENHANCEMENT

HUI LI

PhD

The Hong Kong Polytechnic University

2020

The Hong Kong Polytechnic University
Department of Computing

Improving Image Quality by Dynamic Range and Resolution Enhancement

Hui LI

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

June 2019

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

Hui LI (Name of student)

Abstract

Digital cameras play a vital role in recording various images in our daily life. However, the quality of captured images may not meet our requirements due to the limited dynamic range and resolution of imaging sensors. High dynamic range imaging (HDRI) and super-resolution techniques have been developed to improve the image quality. In this thesis, we investigate some key issues in HDRI and super-resolution. Specifically, we study the problems of tone mapping, multi-exposure fusion (MEF), and real-world single image super-resolution.

Generally, there are two approaches to achieving HDRI: tone mapping for high dynamic range (HDR) data and multi-exposure fusion. For the HDR data captured by high-bit sensor, a process called tone mapping is needed to display the image on conventional low dynamic range display devices. We present a novel clustering based content and colour adaptive tone mapping method. First, the radiance map containing HDR contents is partitioned into various clusters via clustering. Then a Principal Component Analysis (PCA) dictionary is learned for each cluster. For each input patch, it is adaptively assigned to the closest cluster, and projected onto the dictionary associated with this cluster. By adopting an effective compression function to adjust the coefficients, tone mapping can be achieved while some noise and trivial details can be suppressed. To reduce the computational cost, an off-line version of the proposed method is built by pre-learning PCA dictionaries from natural images. The experimental results demonstrate that the proposed tone mapping method can produce high-quality image with well-preserved local contrast, as well as vivid colour appearance with little artefacts.

MEF is another popular way for image dynamic range enhancement. We propose a multi-scale fast MEF approach based on the structural patch decomposition

of images. The proposed method rephrases patch decomposition into image decomposition and merges the weights of signal strength and structure components. As a result, it decomposes each image into two components: one base layer and an implicit detail layer. We indicate that the patch decomposition based aggregation is essentially a process of mean filtering of weight maps, based on which the computational complexity of patch aggregation can be largely reduced so that it is independent of patch size. The multi-scale technique can be implemented by progressively decomposing the base layer, which helps alleviating the hallo effect. The weights at each scale can be designed in a scale-aware manner based on simple image statistical information. Our approach can produce pleasing MEF results with less artefacts and computational cost than previous state-of-the-art methods for both static and dynamic scenes.

We also investigate the MEF methods by deep learning. In particular, we make the first attempt to use deep features to fuse multi-exposure images via an unsupervised method, while the features are extracted via a pre-trained network. We employ the shallow features guided by the deeper semantic features in a classification network to design the fusion weight maps, which are computed via local visibility and temporal consistency. The proposed method works well in both static and dynamic scenarios, bring pleasing fusion results with less computational cost. We then explore an end-to-end network for MEF based on two public datasets. The trained network can effectively fuse multi-exposures images from the test dataset. However, the generalization ability is limited because of limited number of available dynamic multi-exposure images with ground-truth.

In addition to dynamic range, resolution is another important factor affecting image quality. Deep convolutional neural networks (DCNN) have achieved impressive performance in super-resolving bicubicly downsampled low-resolution (LR) images from their high-resolution (HR) counterparts. However, the DCNN models trained by such simulated data become less effective when applied to real-world LR images because the practical degradation of real images is far more complicated than bicubic downsampling. To improve the super-resolution performance of real-world images,

we construct a novel dataset of LR and HR pairs captured by adjusting the lens focus of digital cameras. With the new dataset, a plain regression network with simple loss functions can generate desirable results in real-world image super-resolution. Compared with other DCNN models driven by simulated data, our model can better preserve the fine-scale image edges and textures.

To sum up, in this thesis we proposed a local adaptive tone mapping method, a fast multi-scale patch decomposition MEF method, deep learning based MEF methods, and a new real-world image super-resolution method. The developed methods demonstrated competitive performance to improve the image quality with high efficiency.

Keywords: Image Enhancement, HDRI, Multi-exposure fusion, Clustering, Patch-decomposition, Real Super-resolution, Deep Learning

Publications arising from the thesis

Paper

1. **Li Hui**, Jia Xixi, Zhang Lei, "Clustering based content and color adaptive tone mapping", Computer Vision and Image Understanding, 168, 37-49, 2018.

2. **Li Hui**, Zhang Lei, "Multi-exposure fusion with CNN features", in IEEE International Conference on Image Processing, 1723-1727, 2018.

3. Ma Kede, **Li Hui**, Yong Hongwei, Wang Zhou, Meng Deyu, Zhang Lei, "Robust multi-exposure image fusion: a structural patch decomposition approach", IEEE Transactions on Image Processing, 26(5), 2519-2532, 2017.

4. **Li Hui**, Ma Kede, Yong Hongwei, Zhang Lei, "Multi-scale fast structural patch decomposition for multi-exposure image fusion", submitted to IEEE Transactions on Image Processing.

Patent

1. **Li Hui**, Jianrui Cai, Zhang Lei, "Real-world image super-resolution with long-short focus image dataset", US Patent Application (P20303US00).

Acknowledgements

I want to take this opportunity to thank all the people who have given me help, care, inspiration, and support during my PH.D. study.

I want to show my greatest gratitude to my supervisor, Prof. Lei Zhang for his patient guidance and suggestions on my research. When I encounter some problems in my research, the discussions with him can efficiently help solve the problems. The passion, endeavor and strict standard he shows motivate me to conduct good research. The connection with DJI company provided by Prof. Zhang is very helpful to enrich my experience and knowledge about industry.

Second, I would like to thank all the colleagues in my Prof. Zhang's team and my office PQ503 for extending my horizons through wide discussions about different research topics. I also thank the friends I made for the wonderful get-togethers and memories.

Lastly, I am deeply grateful to my parents for their selfless love, as well as my dear wife Jia Liu who helps me get through the hard time via incredible support and encouragement.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

Because of the constraints of environmental conditions, imaging devices and display devices, the captured digital images are generally not desirable in visual quality. The dynamic range and resolution are two main factors. It is costly to overcome these problems with a better camera sensor that has a higher resolution or bit-depth. An alternative and more cost-effective approach is to employ techniques of image enhancement to improve resolution or dynamic range. Image enhancement plays a critical role in the fields of computer vision, computational photography, and image processing. It can effectively improve the image visual quality without adding increasing imaging hardware cost. The goal of this thesis is to improve the image quality in terms of the dynamic range and resolution by utilizing high dynamic range imaging (HDRI) and super-resolution techniques. In this chapter, we give a brief introduction on HDRI and super-resolution, some baseline methods, existing limitations, and our contributions.

1.1 High dynamic range imaging

HDRI has been an important topic in the field of computer vision and computational photography. Dynamic range refers to the ratio of maximum to minimum irradiance in a natural scene. The dynamic range of a natural scene is usually very high,

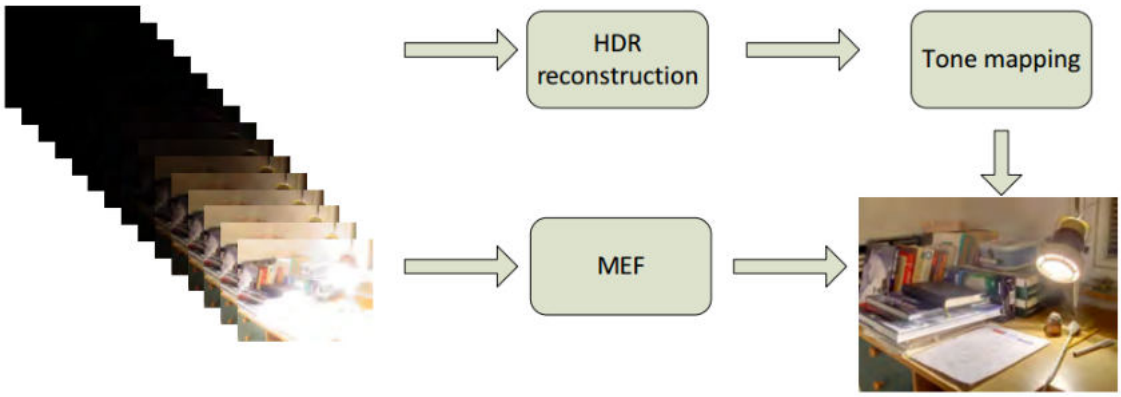


Figure 1.1: High dynamic range imaging framework with multi-exposure image sequence.

approximately 14 orders of magnitude [98, 16]. However, due to the low dynamic range (LDR) of current sensors, under-/over-exposure occurs frequently in everyday photo-taking experiences, leading to unpleasing information loss.

Faithful reproduction of natural scenes with high dynamic ranges is a quite challenging task [98]. A direct way to obtain HDR content is to record the scene using advanced imaging systems. Specialized HDR camera systems [101, 2] have been designed to improve the capability of measuring light on single camera sensor or split light onto multiple sensors with varying exposure. This strategy is not widely accessible due to the complexity of manufacturing new HDR camera hardware.

Another broadly used strategy to extend the camera dynamic range is to take a sequence of images under different exposure levels via exposure bracketing [75, 79, 120]. With this multi-exposure sequence, there are two categories as shown in Fig. 1.1 of approaches to obtain the HDR-like images: multi-exposure image fusion (MEF) [120] in image domain, and HDR content reconstruction in radiance domain [10, 84, 4] and tone mapping for displaying on LDR devices.

The approach to reconstruct HDR content in radiance domain needs to recover the camera response function (CRF) [10, 84, 26, 50, 61, 4]. When taking a nat-

ural scene, the luminance value is recorded by camera sensor, and then undergos in-camera imaging process which includes a series of non-linear processing such as photoelectric conversion, and analog-to-digital conversion, white balance, tone and gamut mapping, as well as sharpening, etc.

Researchers establish an end-to-end relationship between irradiance and pixel value by denoting the whole middle process as the camera response function. The inverse camera response function can convert pixel values to irradiance, from non-linear space to linear space. The imaging relationship can be formulated as:

$$Z_{i,j} = f(H_{i,j}) = f(E_{i,j}\Delta t_j) \tag{1.1}$$

where E, Z indicate the irradiance and pixel value at the pixel (i) on a camera sensor in the $j-th$ exposure image with exposure time Δt_j, respectively. f is the camera response curve.

Since f is a monotonic function, we impose inverse transform on equation 1.1 to get:

$$f^{-1}Z_{i,j} = E_{i,j}\Delta t_j \tag{1.2}$$

With the logarithm transform, we have:

$$g(Z_{i,j}) = lnf^{-1}(Z_{i,j}) = lnE_{i,j} + ln\Delta t_j \tag{1.3}$$

where we have $g = lnf^{-1}$.

The minimum least square objective function F is formulated as:

$$F = \sum_{i=1}^{N}\sum_{j=1}^{P}(g(Z_{i,j}) - lnE_{i,j} - ln\Delta t_j)^2 + \lambda\sum_{z=1}^{254}(w(z)g^{''}(z))^2 \tag{1.4}$$

where N is the pixel indexes at each image. P is the number of images. $g^{''}$ is the discrete second-order derivative of g used to constrain the smoothness of the recovered radiometrical response. The parameter λ controls the smoothing degree.

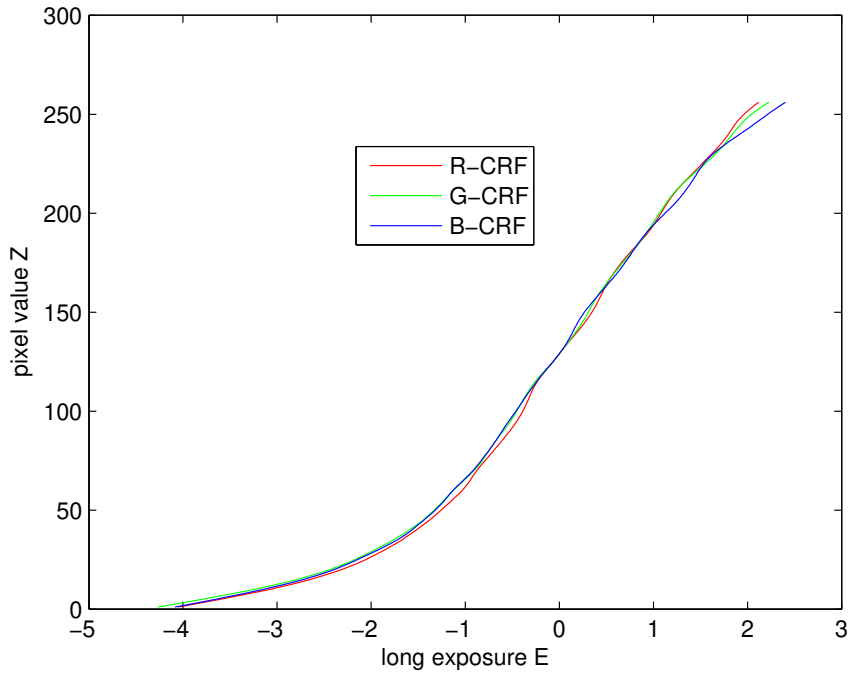


Figure 1.2: Recovered camera response function of R, G, B channel via above image sequences.

Since the pixel value around 0 and 255 is not stable, we add a weighting function as the constraint.

$$w(z) = \begin{cases} z, & z \leq 128 \\ 255 - z, & z > 128 \end{cases} \tag{1.5}$$

By minimizing the objective cost, we can get the closed-form solution of g via singular value decomposition. With the help of the g curve, we can project the pixel value into radiance domain. Weighting the irradiance in radiance domain will result in the final HDR data:

$$lnE_i = \frac{\sum_j w(Z_{i,j})(g(Z_{i,j} - ln\Delta t_j))}{\sum_j w(Z_{i,j})} \tag{1.6}$$

An example calculating the camera response curve is given in Fig. 1.2 by processing a sequence of multi-exposure images with different shutter speeds using the

Figure 1.3: Multi-exposure images with the shutter speed ranging from 1/4000 to 15 seconds.

method mentioned above. The camera response curve is plotted by each channel. Fig. 1.3 shows the sequence of exposure images by adjusting shutter speech with fixed ISO and aperture. The exposure value is affected by three factors: shutter speed, ISO and aperture. During the acquisition of exposure sequences for establishing camera response curve, we control exposure values by adopting different shutters in line with equation 1.1.

There are also other methods estimating CRF. Mitsunaga *et al.* [84] assumed the camera response curve as a polynomial form. The strongness of the assumption does not rely on the specific shutter speed which is difficult to capture for previous low-end cameras, but on the ratio of exposure between frames. The algorithm can

Figure 1.4: An example of tone mapping.

recover camera response curve via only a few exposures, but the exposures have to be captured in static scene. Lee *et al.* [61] calculated the camera response by adding rank-1 constraint on a group of vectorized exposures. Badki *et al.* [4] extended the algorithm to dynamic situation by modifying the objective function based on Lee's model [61]. Badki's work can also be explained as a deghosting method taking advantage of low-rank property.

1.2 Tone mapping

The dynamic range human visual system can perceive is much lower than that of the scene. But with the adaptive adaption ability of human eyes, humans can feel relatively high dynamic range at least five orders of magnitudes. With the high-bit HDR image available, one important issue is how to display the HDR data as shown in Fig. 1.4. The standard display devices such as LCD, CRT, projectors and printers mostly have a low dynamic range and cannot display HDR images directly. To fill in the gap between HDR data and LDR display, techniques have been developed to compress the dynamic range of HDR data for effective display, which are called tone mapping or tone reproduction [22, 15, 99, 77, 62, 57]. A good tone mapping algorithm should faithfully preserve the image detailed features and colors while reducing the

Figure 1.5: Multi-exposure image fusion in static scenes.

irradiance level. In the past two decades, a number of studies have been conducted to develop effective tone mapping algorithms. Generally speaking, the tone mapping methods fall into two primary categories: global tone mapping methods [15, 115] and local tone mapping methods [22, 99].

1.3 Multi-exposure image fusion

Multi-exposure image fusion [82, 67] shown in Fig. 1.5 provides us with another approach to achieve high dynamic range image. It refers to the fusion in 8-bit pixel domain. This technique has been broadly applied to consuming devices, such smart mobile phone and digital camera. Multi-exposure fusion skill has several advantages over HDR reconstruction and tone mapping. Firstly, there is no need to recover

camera response function, which can reduce error caused by constructing HDR content. Secondly, storing the HDR data whether it is synthetic data or real raw data is a big burden for limited storage in consuming devices. In a world, MEF offers a simpler and more direct alternative by performing fusion in intensity domain, which has been widely employed in mobile devices for HDR imaging [35].

This static scene fusion process can be expressed as follows:

$$X(i,j) = \sum_{k=1}^{K} W_k(i,j) X_k(i,j) \tag{1.7}$$

where the $W_k(i,j)$ and $X_k(i,j)$ indicate the weight and intensity values at the pixel(i,j) in the $k-th$ exposure image, respectively; $X(i,j)$ represents the fused image.

A great deal of research has explored this fusion issue. Multi-exposure fusion can be divided into two categories: static scene fusion [25, 96, 82, 106, 105, 111, 69, 64, 65, 27, 133, 6] and dynamic scene fusion [24, 91, 38, 73, 39, 60, 47, 42, 102, 95, 90]. In the following section, we provide an overview of existing MEF algorithms with an emphasis on how different methods compute perceptual weights for fusion, and how they design exposure-invariant features for motion estimation.

1.3.1 MEF methods for static scenes

Static MEF methods mainly consist of weight map computation and smoothing in a single-scale or multi-scale fashion [82], followed by post-processing such as detail-enhancement [69]. The weight map smoothing occurs explicitly in pixel-level fusion to keep spatial consistency. Patch-level fusion smooths the weight map implicitly via aggregating overlapping patches [25, 76]. Multi-scale decomposition is widely used in MEF for halo reduction [82, 53]. Post-processing is often adopted to further improve the visual quality of fused images.

The well-known MEF method proposed by Mertens *et al.* [82] computes the weight map via contrast, color saturation, and well-exposedness measurements. Fusion is accomplished in a multi-scale framework where the input images are decomposed into a Laplacian pyramid and the weight maps are smoothed within a Gaussian pyramid. While computationally cheap, this method suffers from possible halo artifacts and detail loss. Li *et al.* enhanced the details of the Mertens' results by solving a quadratic optimization problem in single scale [69] or multi-scale [67]. Shen *et al.* performed MEF in a boosting Laplacian pyramid [104]. Kou *et al.* [53] replaced Gaussian smoothing in [82] with gradient domain guided smoothing to reduce halos. Ancuti *et al.* [1] provided a fast single-scale approximation to [82] by Gaussian filtering the weight map with a larger kernel size and adding back the details extracted using a second-order Laplacian filter.

Li *et al.* [65] decomposed the input sequence into a base layer and a detail layer, whose weight maps were computed by saliency measurements and refined by guided filters [36] with different parameters. Raman and Chaudhuri [96] directly adopted the detail layer as the weight map, which results in somewhat dreary appearance. Goshtasby [25] designed the weight map based on the max-entropy principle and smoothed it with a monotonic blending function to reduce blocking artifacts.

Optimization-based methods have also been used in MEF. Ma *et al.* [74] employed a gradient descent-based method to optimize MEF-SSIM [78] in the image space. Despite visual quality improvement, their algorithm is prohibitively slow. Prabhakar *et al.* [94] trained a feed-forward convolutional network by optimizing MEF-SSIM. The method works reasonably well on extreme situations, but it is not flexible to handle sequences of arbitrary number of exposures. Cai *et al.* [7] made use of 13 existing MEF methods to generate a set of fused candidate images, and manually picked the best ones as the ground truths to train a convolutional network for single image contrast enhancement. Since this process requires extensive human

interventions, the resulting number of sequences for training is quite limited, which may hinder the generalizability of the learned network.

1.3.2 MEF methods for dynamic scenes

The fusion methods reviewed above were specially designed only for static scenes. Once it is applied in the dynamic scene where some objects are moving or the camera is shaking, ghosting artefact will occur as shown in Fig. 1.6. In the situation, the head of that horse is moving across the scene during photography, leading to ghosting appearance in the final fusion result. The reason of generating ghosting effect lies in two aspects: the motion of the camera as well as the moving objects. In order to eliminate the ghosting due to camera moving, one way is to put the camera on a tripod. Another solution for addressing this kind of ghosting is to globally register the source exposure images via registration operator such as SIFT, Harris, SURF, MTB. In terms of whether a reference image is selected, there exist two sorts of deghosting strategies. Selecting a reference image such as [70, 39] means that the motion in the reference image is retained while other motions in non-reference images are discarded. By contrast, all motions are removed without choosing a reference image [24, 133, 73].

Dynamic scene fusion methods fall into two categories in terms of deghosting: radiance domain based, and image domain based. In the intensity domain, Kang first boosted the intensity values of adjacent frames in order to compensate for the exposure changes and then ran Lucas-Kanade method to compute optical flow, which is refined by a hierarchical homography if necessary [46]. Khan used a kernel density estimation scheme to determine the probability that a pixel belongs to the background [47]. However, this method fails to deal with small random motion such as ripples and tree branches in the wind. Jacobs detected pixels that may belong to moving objects using entropy measure with the assumption that entropy is invari-

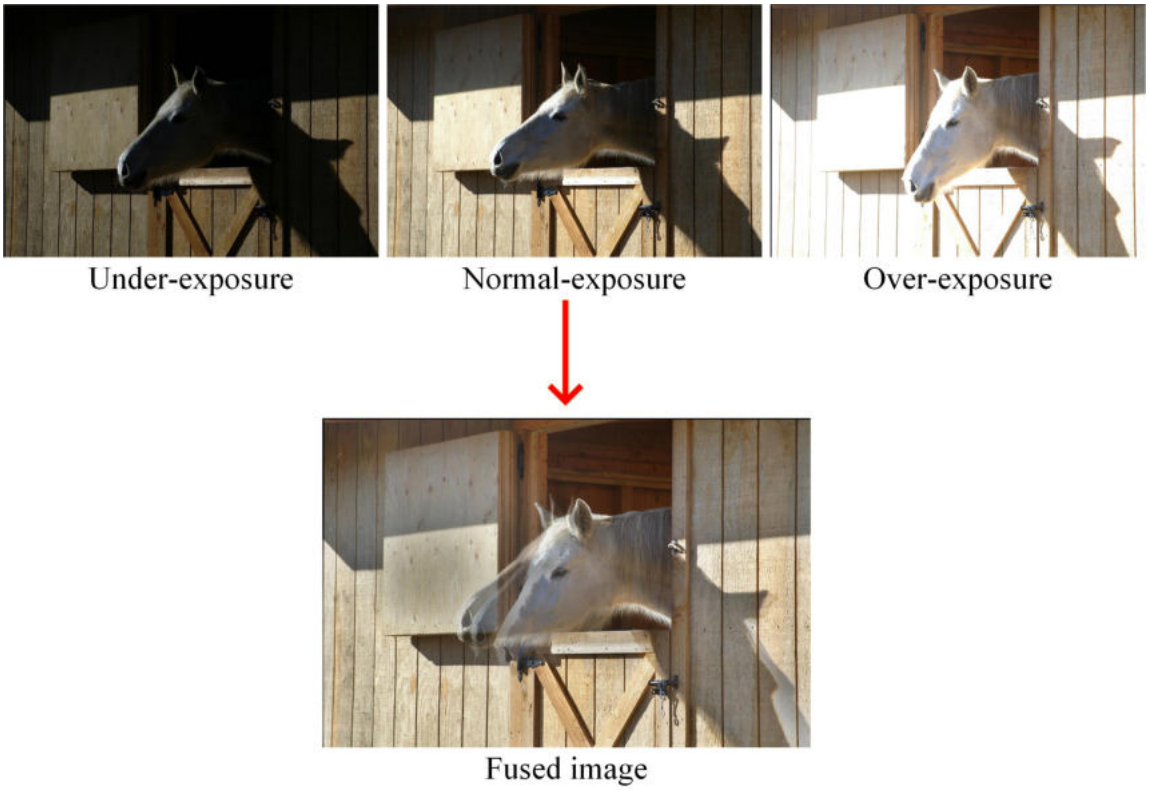


Figure 1.6: An example of ghosting effect with multi-exposure image fusion in dynamic scenes.

ant under an injective function [42]. Median threshold bitmap is adopted in [91] to detect the motion pixel and to select the best available exposure for fusion. Similar to the method in [18], intensity information in only one exposure is used in some local regions, which may be limited in expanding the dynamic range of those particular scene areas and may also cause local luminance inconsistency. In the radiance domain, Eden recovered the HDR image by deliberately setting each radiance value from one of the input images, which may cause moving object duplication or deformation [18]. By exploiting the linearity between sensor radiance and exposure time, Gallo checked the inconsistent patches from other exposures w.r.t. the chosen reference patch and blended consistent patches in the gradient domain to avoid visual block artifacts caused by inaccurate CRF estimation [24]. Adopting a bidirectional

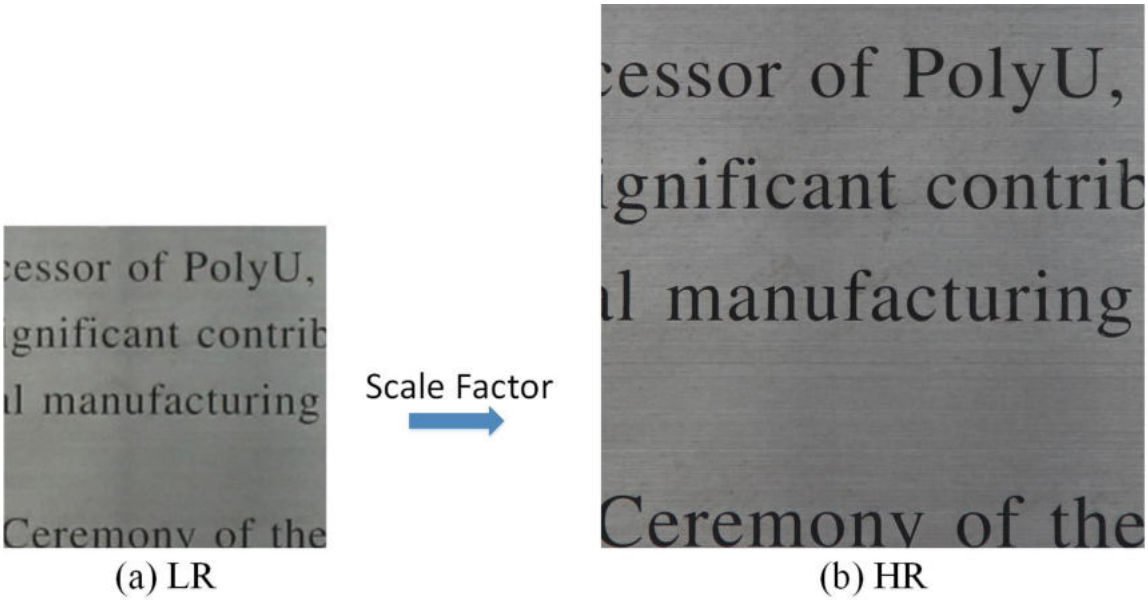


Figure 1.7: An example of real image super-resolution.

similarity measure [109], Sen tackled camera and object motion together in a patch-based energy minimization framework [102]. Lee estimated a binary ghost indication matrix via ranking minimization with sparsity and connectivity constraints as well as prior information on under- and over-exposed regions [60]. Oh *et al.* also exploited the low rank property of the source image sequence using partial sum minimization of singular values and extended it to matrix completion to account for complex motion, which however requires human interactions for moving object inclusion [90].

1.4 Real image super-resolution

In addition to the dynamic range, resolution is another important factor which has a big impact on image quality. In this thesis, we aim to address the issue of real single image super-resolution (SISR). Real-world super-resolution is to directly super-resolve an image without prior downsampling as shown in Fig. 1.7. SISR aims to recover a high-resolution (HR) image from a degraded high-resolution (LR) image, which can effectively overcome the resolution limitation of low-cost imaging sensors

or enhance existing images. As a classic inverse problem, it can be formulated as:

$$y = DHx + n \tag{1.8}$$

where x is the HR image to be recovered, y is the LR image. H, D and n are the blur kernel, down-sampling operator, and additive noise, respectively. In general, n is assumed free or additive white Gaussian noise (AGWN), and H is an identity matrix.

Various works have been proposed in the past decades. Early studies mainly focused on analytical interpolation methods such as nearest, bilinear and bicubic interpolation kernel due to limited computational resource. Despite fast implementation, these methods suffer from severe edge and detail loss in the zoomed HR image. Recent research works can be classified into model-based optimization methods and learning based methods. It is a severely ill-posed problem to recover x from y. In the model-based methods, some prior information such as sparsity [123, 31] and non-local similarity [13] need to be used to better estimate the missing pixels especially around sharp edges. The main deficiency of this kind is the high computational cost and complex parameter adjustment.

Recently, CNN has been successfully employed in image super-resolution [11], obtaining state-of-the-art performance in terms of signal-to-noise ratio (PSNR) because of its powerful discriminative learning ability with the help of efficient parallel computing. Although these finely designed modes can obtain high PSNR and visual quality in testing images downsampled by bicubic approach, they do not work well in practical applications, where an LR image is amplified directly without pre-bicubic downsampling.

1.5 Contributions and organization

In this thesis, we investigate three key issues in dynamic range and resolution enhancement tasks: tone mapping, multi-exposure fusion, and real-world image super-resolution. The thesis consists of four main research works conducted during my PH.D. study, which are described with details in the following chapters:

In Chapter 2, we propose a clustering based adaptive tone mapping method, which uses non-local redundancy and local statistics for adaptive tone mapping. The tone mapping is implemented on each patch which is decomposed into three components: patch mean, color variation and color structure. The similar structure component is grouped via clustering. The detail structure is projected on the corresponding dictionary constructed via PCA transform. By adjusting the coefficients through an effective s shape curve, the dynamic range is compressed and adjusted. Furthermore, the method can suppress the noise via a hard threshold shrinkage of small projection coefficients. The multi-scale technique is used to reduce halo artefact. The off-line version via pre-training external data is implemented for a reduced computational cost. Experiments have been extended on a large amount of HDR data consisting of synthetic data or HDR raw data. Qualitative and quantitative analysis indicate the effectiveness of robustness of the proposed method.

In Chapter 3, we propose a fast multi-scale multi-exposure image fusion method, which can work well in both static and dynamic scenes. It can be regarded as an extension of a rephrased SPD-MEF. The relationship with classic two-layer decomposition based methods is analysed. The images are decomposed and fused simultaneously at each scale. The high frequency weight is inherited from SPD-MEF, while the low frequency weight is designed by a new well-exposedness measure. The final result is obtained by summing up the fused base layer and various detail layers via upsampling. The multi-scale technique can generate favorable results with

reduced halo effect, faithful color and structure preservation, and decent global contrast. The complexity is linear independent with filter size, which can be applicable in consuming devices.

In Chapter 4, we use deep learning for MEF. A novel CNN feature based MEF method is proposed. When fusing multi-exposure image fusion, we hope to select the good-exposure pixels, and then blend them. The CNN feature can help achieve this target by extracting exposure-aware features. The L_1 norm of the feature vector can reflect the importance of one pixel in a local region. The normalized CNN feature can be used to handle the ghosting artefact in dynamic scene. Additionally, we found that shallow layer feature can be more effective for deghosting than the deep feature. Because shallow layer feature mainly includes edge, gradient or structural information which are exposure-insensitive for the motion detection across the scene. Besides, we use two available datasets to explore an end-to-end MEF methods, which could present decent performance. But the lack of ground-truth in dynamic scene results in ghosting effect.

In Chapter 5, we made the first attempt to address the issue of real-world image super-resolution by establishing long-short focus image dataset by use of four different camera lens. Image registrations based SIFT are employed to crop the HR and LR pairs. The baseline networks using our dataset achieved better results compared with state-of-the-art methods trained by simulated data. The dataset taken by real cameras fits more degradation types than conventional single or multiple degradation assumptions. We also design a hybrid loss to keep a balance between detail preservation and artefact suppression. The non-reference real image super-resolution using general images from six dataset indicate our method shows favorable visual quality with both good edge and texture preservation.

The core works form Chapter 2 to 5 correspond to the Paper 1, 2, 4 and Patent 1.

Chapter 2

Clustering Based Content and Color Adaptive Tone Mapping

Retinex theory has been widely adopted for tone mapping to visualize high dynamic range (HDR) images on low dynamic range display devices by extracting image luminance channel and separating it into a base layer and a detail layer. Many edge-preservation filtering techniques have been proposed to approximate the base layer for Retinex image decomposition; however, the associated tone mapping methods are prone to halo artifacts and false colors because filtering methods are limited in adapting the complex image local structures. We present a statistical clustering based tone mapping method which can more faithfully adapt image local content and colors. We decompose each color patch of the HDR image into three components, *patch mean*, *color variation* and *color structure*, and cluster the patches into a number of clusters. For each cluster, an adaptive subspace can be easily learned by principal component analysis, via which the patches are transformed into a more compact domain for effective tone mapping. Comparing with the popular edge-preservation filtering methods, the proposed clustering based method can better adapt to image local structures and colors by exploiting the image global redundancy. Our experimental results demonstrate that it can produce high-quality image with well-preserved local contrast and vivid color appearance. Furthermore, the proposed

method can be extended to multi-scale for more faithful texture preservation, and off-line subspace learning for efficient implementation.

2.1 Introduction

Given that nowadays most available display devices are 8-bits, the tone mapping operation is needed to reproduce the HDR data on the 8-bit devices for display. In the past two decades, a number of studies concerning tone mapping algorithm have been conducted.

Due to the limited computational resources, early studies [15, 115, 118, 58] focus on designing simple global tone mapping operators. Tumblin et al. [115] proposed a non-linear tone mapping algorithm according to the brightness perception of human visual system. Ward et al. [118] compressed image contrast instead of absolute luminance using a simple linear compression function. Larson et al. [58] applied histogram adjustment to tone mapping by preserving the histogram distribution of the original HDR data. The adaptive logarithmic mapping in [15] compresses the dynamic range with different logarithmic bases. The higher irradiance is compressed via log_2, whereas the lower irradiance via log_{10}, to achieve desirable contrast and detail preserving. Reinhard et al. [97] proposed a simple and practical s curve for global tone mapping in independent channels. The global operators are computationally efficient without halo artifacts. However, the local contrast and visibility of details in the produced LDR images are not satisfactory.

Recent studies focus more on local tone mapping techniques. Fattal et al. [22] designed a novel local tone mapping operator based on gradient attenuation. They compressed the drastic irradiance changes by reducing the large gradients under a multi-scale framework. Reinhard et al. [99] classified the dynamic range of display devices into 11 zones according to the different irradiance in HDR data. Li et

al. [66] put forward a multi-resolution image decomposition method using symmetrical analysis-synthesis filter banks for local tone mapping. The gain map of each subband is calculated to alleviate the halo artifacts. Shan et al. [103] developed a globally local optimization method with a locally linear model, where the guidance map is constructed via local statistical information. Gu et al. [29] replaced the linear assumption [103] with the local non-linear gamma correction. Ma et al. [77] designed a tone mapped image quality index (TMQI) and performed dynamic range compression by optimizing this index. Chen et al. [9] segmented the HDR image into different regions via the earth mover's distance (EMD), and applied local tone mapping operation on each component. Ferradans et al. [23] proposed a two-stage tone mapping method: human visual system based global tone mapping, followed by optimization based local contrast enhancement. Duan et al. [16] improved the tone mapping performance of [58] by applying adaptive local histogram adjustment on non-overlapped blocks. In general, local tone mapping methods are spatially adaptive, and can reproduce the local details and contrast well. However, these local operators have higher computational cost and are prone to producing halo artifacts [66] and ringing effect [108].

In recent years, researchers have been focusing on the design of various edge-preserving filters for tone mapping. The main principle is to decompose an HDR image into a detail layer and a base layer, and impose different operations on the two layers. In particular, the base layer image can be obtained by filtering the HDR data. Tumblinand et al. [116] made the first attempt to design edge-preserving filters by using anisotropic diffusion to replace Gaussian filtering based on the Retinex theory [43]. Durand et al. [17] developed a fast implementation of bilateral filtering for tone mapping, which can efficiently generate smoothed images while preserving the edges. Based on this framework, many subsequent works [21, 32, 122, 36, 68, 72, 51] have been proposed to better remap the HDR data. In [21], a weighted least

squares based global optimization method was proposed to smooth the HDR data, where a larger weight is given to local details and contours, while a smaller weight is distributed to strong edges. An iterative method was proposed in [32] to improve the solving of weighted least squares. By minimizing the global gradient of an HDR image, Xu et al. [122] used the l_0 norm as the regularizer to smooth the HDR image. He et al. [36] proposed a guided filtering based method for edge preservation. A linear relationship is assumed between the guided image and the image to be filtered to avoid large edge loss. Some works [68, 72, 51] introduce the gradient information as the weight to balance the data term and regularizer term in a local window, which share the similar idea to global weighted least squares.

The luminance edge-preservation filtering based tone mapping algorithms mentioned above can improve the visual quality of tone mapped image; however, the nonlinear filters used by them are not flexible and adaptive enough to fit the various edges and structures in natural images, resulting in halo artifact and false colors. Different from those luminance filtering based methods, in this chapter we develop a statistical clustering based tone mapping method to more effectively exploit the image local and global redundancy. We do not separate an image into luminance and chrominance channels to process; instead, we work on image patches, and decompose a color patch into three components: patch mean, color variation and color structure. It is well-known that there exist repetitive patterns/structures in natural images [130, 14]. Based on the color structure component, we group similar patches into clusters, and use statistical signal processing tools such as principal component analysis (PCA) to define a subspace of the patches in a cluster. Consequently, we can project each patch into a more compact domain, where the tone mapping operation can be more effectively performed. Compared with the edge-preservation filtering based methods, our proposed statistical clustering based method is more local content and color adaptive and robust since it exploits the image global redundancy to

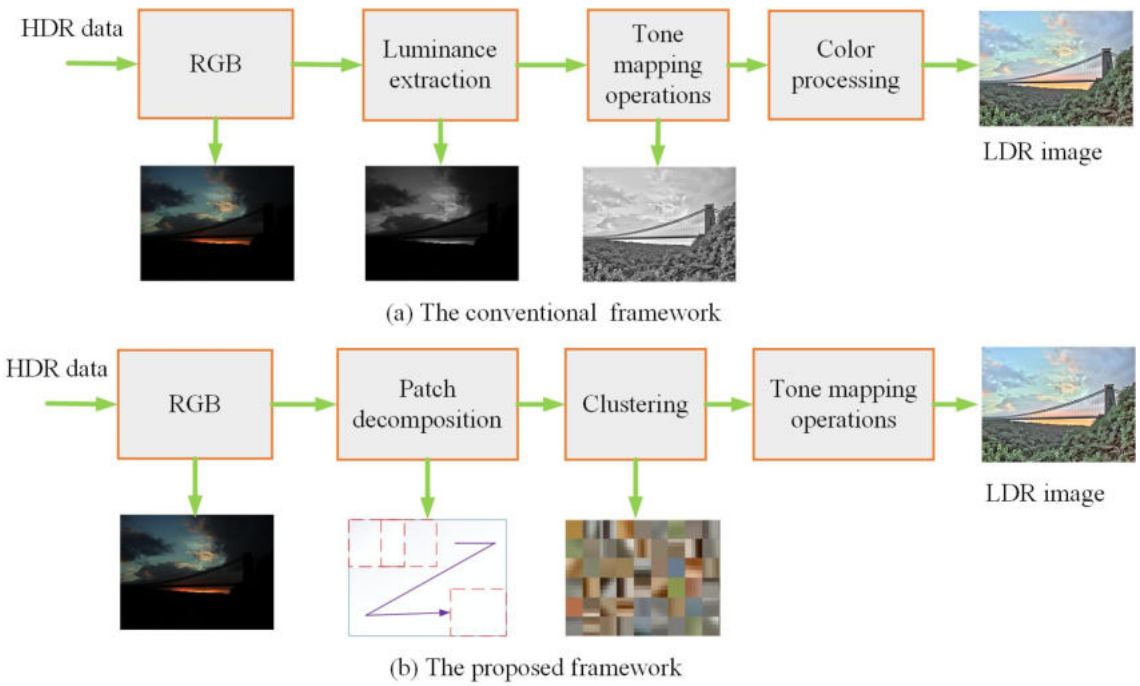


Figure 2.1: (a) The traditional tone mapping framework and (b) our proposed framework.

decompose local structures.

The main contributions of this chapter lie in the following aspects. 1) Instead of using the deterministic edge-preserving filters, we leverage statistical clustering methods to better represent the local color structures of HDR images. Each patch will be adaptively processed based on its cluster. 2) We perform tone mapping in the PCA transformed domain other than the intensity domain, where the coefficients have explicit physical meanings and can be more effectively compressed. 3) Different from previous methods which extract luminance channel and perform layer separation on it, we do not extract luminance channel but process image luminance and chrominance information simultaneously.

The rest of this chapter is organized as follows. Section 2 presents the proposed method in detail. Section 3 presents extensive experimental results and discussions. Section 4 concludes the chapter.

2.2 The proposed tone mapping framework

Most previous tone mapping methods process luminance and chrominance separately. A typical framework of conventional tone mapping methods is shown in Fig. 2.1(a). Given an HDR image in RGB format, the luminance channel is first extracted as $L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B$ for the XYZ color space [22], or $L = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$ for the YUV color space [68]. In some literature [28], the average of R, G, B channels $L = 1/3(R + G + B)$ is employed as the luminance. After dynamic range compression on luminance, the chrominance is processed based on the compressed luminance to reproduce the tone mapped image. The widely used color processing operation is $C_{out} = (\frac{C_{in}}{L_{in}})^s \cdot L_{out}$, where C represents the chrominance channel, L_{in} and L_{out} denote the luminance before and after HDR processing, and s adjusts the color saturation of the tone mapped image. The empirical value of s is between 0.5 and 0.9 [28].

In our proposed method, we do not separate image into luminance and chrominance channels to process. Instead, we propose a very different approach, whose framework is shown in Fig. 2.1(b). We partition the input RGB image into overlapped color patches, and decompose each patch into three nearly uncorrelated components. The color patches are clustered into a number of clusters, and statistical analysis is used to compress each HDR patch to an LDR one. The flowchart of the proposed method is shown in Fig. 2.2. The main procedures of the proposed method include: logarithmic transform, patch decomposition, clustering and PCA transform, range adjustment, patch reconstruction, aggregation and post-processing. The details of the proposed method are presented in the following.

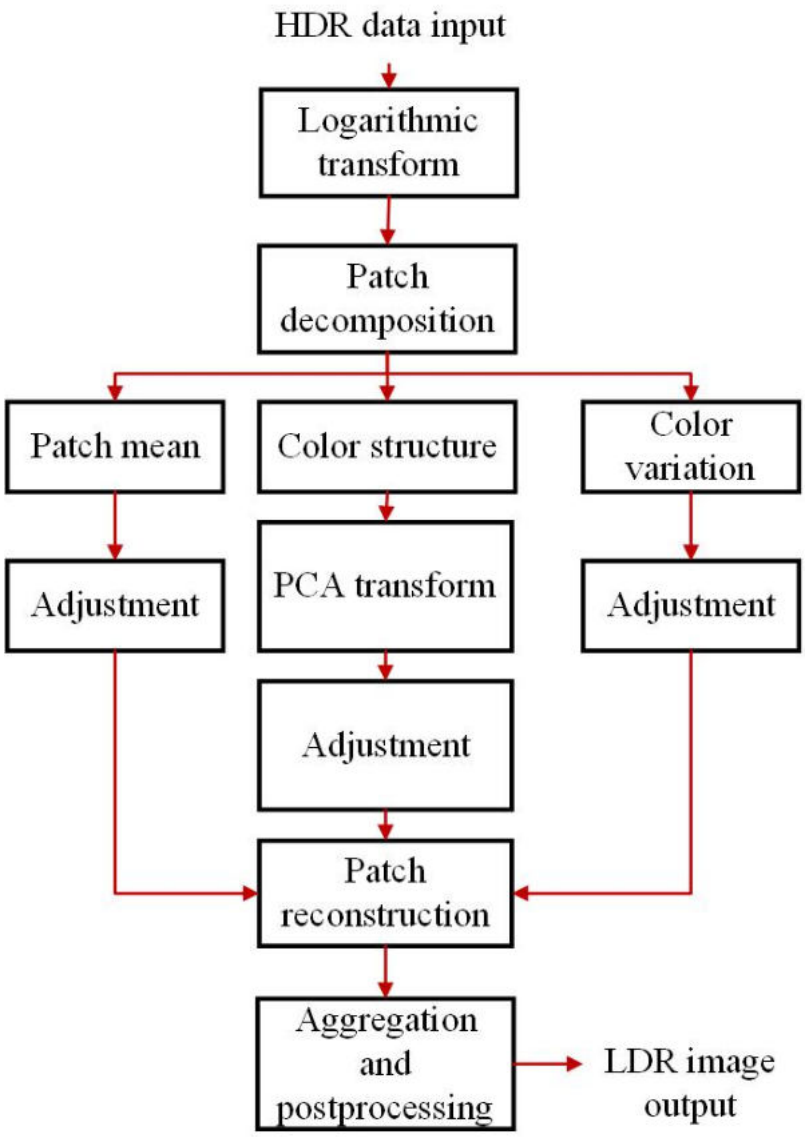


Figure 2.2: Flow chart of the proposed tone mapping method.

2.2.1 Patch decomposition

Like in many existing tone mapping methods [16, 28, 83], our method needs a simple global tone curve for initialization. Considering the characteristics of human visual system, the logarithmic function is used to this end:

$$\mathbf{L}(i,j,c) = log(\mathbf{I}(i,j,c) \cdot 10^6 + 1) \tag{2.1}$$

where $\mathbf{I}$ is the input HDR image, (i, j) refers to the spatial location, and $c \in \{r, g, b\}$ represents the R, G, and B channels. We then apply patch decomposition to $\mathbf{L}$. We partition the HDR image $\mathbf{L}$ into many overlapped patches (e.g., of size 7×7) with stride q (e.g., $q = 2$ in our implementation). Denoted by $\mathbf{x}$ an extracted color patch and by $\mathbf{x}_c$ the patch in channel R, G or B. The local mean of each channel $\mathbf{x}_c$, denoted by m_c, is calculated by averaging all pixels in $\mathbf{x}_c$. We then subtract the mean from $\mathbf{x}_c$:

$$\bar{\mathbf{x}}_c = \mathbf{x_c} - \mathbf{1} \cdot m_c \tag{2.2}$$

where $\mathbf{1}$ is a vector with all elements being 1 and it has the same size as $\mathbf{x}_c$. One can see that $\bar{\mathbf{x}}_c$ contains the direct current (DC) removed detail structure of $\mathbf{x}_c$.

The mean m_c is a scalar representing the DC amount of patch $\mathbf{x}$ in channel c. The variation of m_c across channels can reflect the color appearance in that patch. For example, if all the three values of m_c are the same, that patch will be a gray level patch. We can calculate the color variation across channels as:

$$\bar{m}_c = m_c - m \tag{2.3}$$

where $m = (m_r + m_g + m_b)/3$ is the average of the three m_c. Clearly, m is the average of all pixels in the color patch $\mathbf{x}$.

With the m, $\bar{m}_c$, and $\bar{\mathbf{x}}_c$ defined above, for each patch we can decompose it into three components:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} \bar{\mathbf{x}}_r \\ \bar{\mathbf{x}}_g \\ \bar{\mathbf{x}}_b \end{bmatrix} + \begin{bmatrix} \mathbf{1} \cdot m_r \\ \mathbf{1} \cdot m_g \\ \mathbf{1} \cdot m_b \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \cdot m \\ &= \bar{\mathbf{x}} + \bar{\mathbf{m}} + [\mathbf{1}; \mathbf{1}; \mathbf{1}] \cdot m \end{aligned} \tag{2.4}$$

We call the 1st component $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_r; \bar{\mathbf{x}}_g; \bar{\mathbf{x}}_b]$ the color structure since it preserved the detailed local structural information in the three channels, the 2nd component $\bar{\mathbf{m}} = [\mathbf{1} \cdot m_r; \mathbf{1} \cdot m_g; \mathbf{1} \cdot m_b]$ the color variation since it reflects the color differences across three channels, and the 3rd component m the patch mean since it is the mean value of all pixels in the three channels.

2.2.2 Clustering and PCA transform learning

Given an input HDR image, a large number of patches $\mathbf{x}$ will be extracted. For example, we extract 185754 7×7 patches with stride 2 for an image of size 1000×750. It has been widely accepted that there will be many patches sharing a similar structure in an image [130, 14, 121]. After removing the DC component, some patches with different intensity levels may also have similar structure. Therefore, we can cluster the patches into different clusters based on the color structure component $\bar{\mathbf{x}}$. The classical clustering methods such as K-means [130] and Gaussian Mixture Model (GMM) [121] can be used to this end. We choose K-means because it has much lower computational cost while leading to similar tone mapping results to GMM based on our experiments. We stretch each $\bar{\mathbf{x}}$ to a vector, and apply K-means clustering to the vectorized color structure components $\bar{\mathbf{x}}$ (note that $\bar{\mathbf{x}}$ contains the detailed features from all the R, G and B channels). Suppose that K clusters are obtained. For each cluster, we calculate the covariance matrix of the vectors $\bar{\mathbf{x}}$ within it, denoted by $\mathbf{\Phi}$. Since the covariance matrix $\mathbf{\Phi}$ is positive semidefinite, we can have its eigenvalue decomposition as:

$$\mathbf{\Phi} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \tag{2.5}$$

where the orthogonal matrix $\mathbf{Q}$ is composed of the eigenvectors of $\mathbf{\Phi}$. The so-called principal component analysis (PCA) transform matrix can be easily obtained as [130]:

$$\mathbf{P} = \mathbf{Q}^T \tag{2.6}$$

Since the patches in one cluster are similar in structure, the eigenvectors associated with the first a few largest eigenvalues will be able to represent the most important common structures in that cluster (i.e., the principal components). With the PCA transform matrix $\mathbf{P}$, for each patch $\bar{\mathbf{x}}$ within that cluster, we can transform it into the PCA domain as:

$$\bar{\mathbf{y}} = \mathbf{P}\bar{\mathbf{x}} \tag{2.7}$$

Note that the coefficients in $\bar{\mathbf{y}}$ will be much sparser than those in $\bar{\mathbf{x}}$. The small coefficients correspond to noise interference and trivial structures. The modest coefficients correspond to image fine-scale details. The large coefficients correspond to image principle structures. Usually, only the first a few coefficients in $\bar{\mathbf{y}}$ will be significant, while the remaining being close to zero. Therefore, compressing the dynamic range of $\bar{\mathbf{y}}$ will be much easier and more robust than that of $\bar{\mathbf{x}}$. This is one of the essential reasons that why our method works for tone mapping.

2.2.3 Dynamic range adjustment and patch reconstruction

To achieve the tone mapping of patch $\mathbf{x}$, we need to adjust the values of m, $\bar{\mathbf{m}}$, and $\bar{\mathbf{x}}$. For component $\bar{\mathbf{x}}$, we transform it into the PCA domain via Eq. 2.7 and process $\bar{\mathbf{y}}$. The smallest coefficients in $\bar{\mathbf{y}}$ are usually produced by the trivial structures, fluctuations and even noise in $\bar{\mathbf{x}}$, and therefore we first remove them for a more stable tone mapping. Denote by max the maximal absolute value of all coefficients in $\bar{\mathbf{y}}$. Since noise mostly corresponds to the smallest PCA coefficients, a simple empirical threshold is good enough to suppress the noise. In order to keep the details of the original data as much as possible while removing noise, a small threshold is

empirically selected. We set those coefficients whose absolute value is smaller than 0.1 max to 0.

For the task of tone mapping, the large PCA coefficients (corresponding to image large scale structures) in $\bar{\mathbf{y}}$ should be compressed, while the smaller coefficients (corresponding to image fine scale textures) should be maintained or enhanced slightly. To this end, an s-shaped curve could be employed to adjust the coefficients. The commonly used s-shaped curves include $arctan$ and $sigmod$ functions. We choose the $arctan$ function to adjust coefficients because it exhibits stronger transition ability in both shadows and highlights, and the adjusting function should be symmetrical to 0 to process the negative coefficients in the PCA transform domain. With the $arctan$ function, we adjust the coefficients in $\bar{\mathbf{y}}$ as:

$$\bar{\mathbf{y}}_a = (1.6/\pi) \cdot arctan(a \cdot \bar{\mathbf{y}}) \tag{2.8}$$

where a is a parameter to control the shape of the curve. Some example curves are plotted in Fig. 2.3. One can see that the smaller the a is, the stronger compression effect on $\bar{\mathbf{y}}$ will be.

For the color variation component $\bar{\mathbf{m}}$, we also use the arctan function but with a different parameter to adjust it:

$$\bar{\mathbf{m}}_b = (1.2/\pi) \cdot arctan(b \cdot \bar{\mathbf{m}}) \tag{2.9}$$

where b is the shape parameter. The patch mean component m changes slowly, which can be linearly compressed by multiplying a weight w. After range adjustment on m, $\bar{\mathbf{m}}$ and $\bar{\mathbf{y}}$, the tone mapped patch of $\mathbf{x}$, denoted by $\mathbf{x}_t$, can be reconstructed as

$$\mathbf{x}_t = \mathbf{P}^T \bar{\mathbf{y}}_a + \bar{\mathbf{m}}_b + [\mathbf{1}; \mathbf{1}; \mathbf{1}] w \cdot m \tag{2.10}$$

where w is a scalar ranging from 0 to 1.

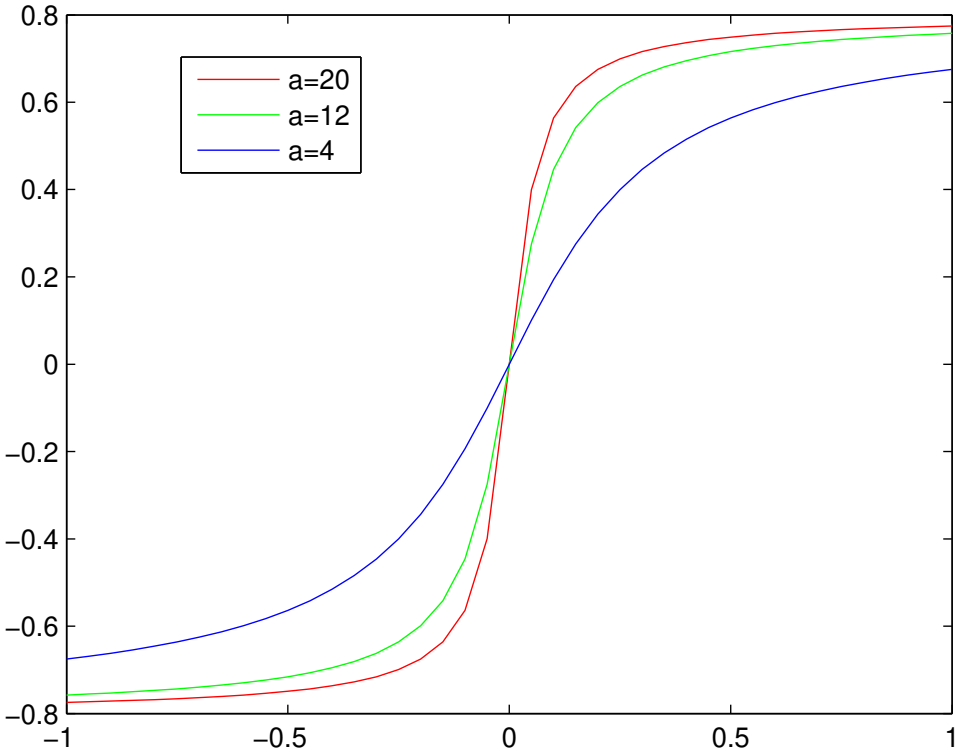


Figure 2.3: The *arctan* function in Equ. 2.8 with different parameters.

2.2.4 Aggregation and post-processing

The operations described in Sections 2.3 and 2.4 are applied to each extracted patch for the input HDR image, and aggregation of the processed patches is needed to reconstruct the tone mapped LDR image. Each tone mapped patch is put back to its original location, while the overlapped pixels in adjacent patches are averaged. In the post-processing stage, the 1% pixels of lowest and highest values are clamped to enhance the primary contrast. Finally, every patch pixel is linearly stretched to $0 - 1$ to fully take advantage of the dynamic range of target display device to show the result.

2.2.5 Extension to multi-scales

In the proposed patch clustering based tone mapping method, each patch will have a mean component (scalar value). The means of all patches will form a smoothed gray level image of the original image. Fig. 2.4 shows an example. Fig. 2.4(a) is the original image (the tone mapped image is shown here for better visibility), and Fig. 2.4(b) is the mean image after patch decomposition. Note that the resolution of mean image is 1/4 of that of the original image because we use a stride factor of 2 (in both horizontal and vertical directions) to extract the patches (size: $7 \times 7 \times 3$).

One can see that there is still certain amount of textures in the mean image. If we compress the mean image by a weight w as shown in Eq.(10), some detailed texture information can be lost in the final tone mapped image. To solve this problem, we could extend the proposed method to multi-scales. More specifically, we extract patches from the mean image, and decompose each patch into two components: patch mean and patch structure. The patch mean is the average of all pixels in a patch, while the patch structure component is obtained by subtracting the mean from the patch. Note that we do not have a color variation component here since the mean image is gray scale. The clustering and PCA transform can then be applied to the patch structure components. By embedding such operations into the framework in Fig. 4.1, we could have a two-scale implementation of the proposed method, which is illustrated in Fig. 2.5.

Our method can be easily extended to more scales by further decomposing the mean image generated on the 2nd scale. Nonetheless, our experiments show that a 2-scale decomposition is enough for most of our test images. In Fig. 2.6(a) and Fig. 2.6(b), we show the single-scale and two-scale tone mapping results by our method. One can see that some detailed structures of the cloud region are lost in the single-scale result image, but they can be preserved in the two-scale result image. In

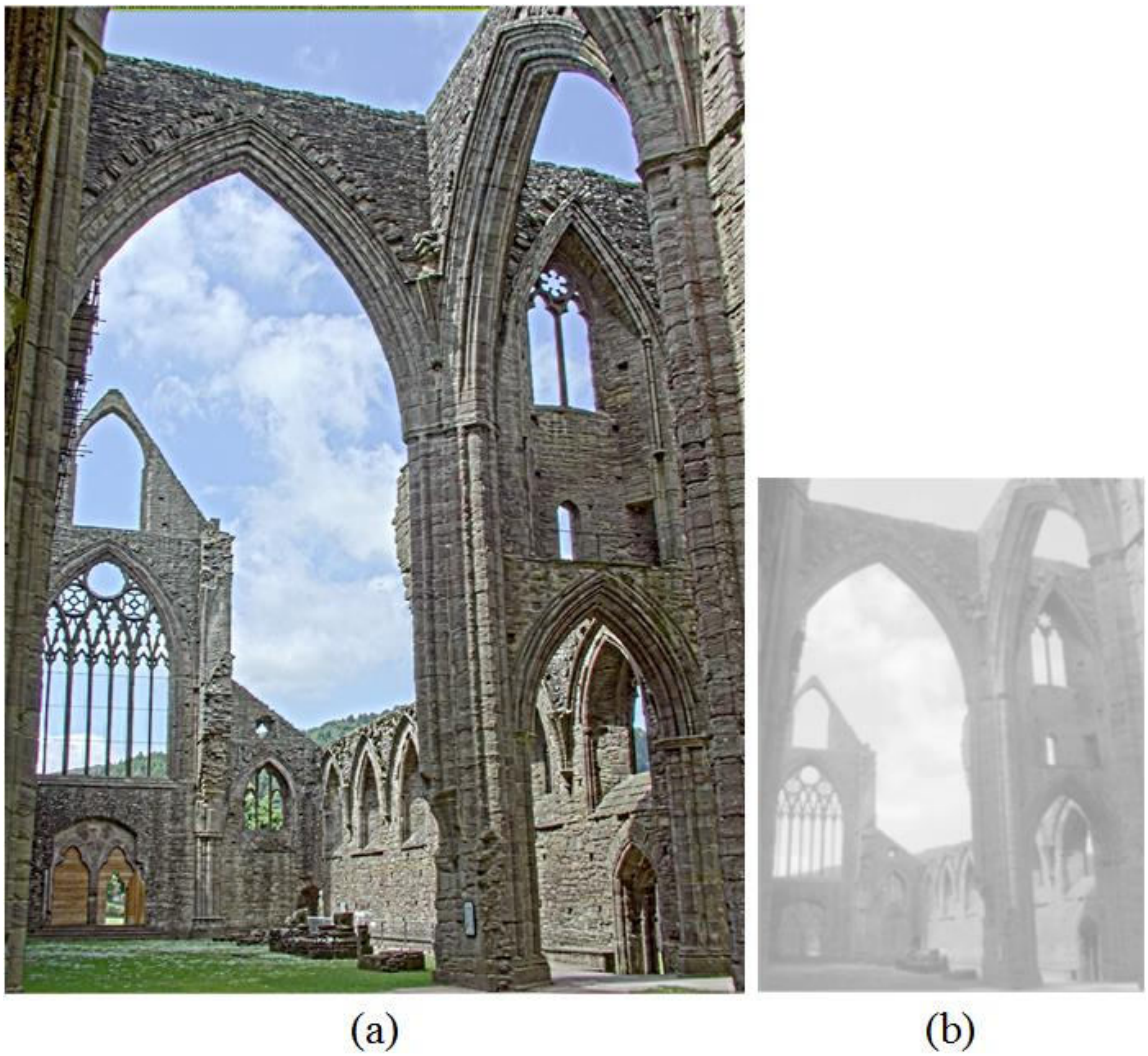


Figure 2.4: (a) The original HDR image (the tone mapped image is shown here for better visibility). (b) The mean image formed by the patch means.

addition, since the mean image is gray scale and has a lower resolution, the two-scale decomposition scheme has similar implementation time to the single-scale scheme.

2.2.6 Offline PCA transform learning

The color structure clustering step is the most time-consuming part in our proposed method. With the K-means clustering algorithm, it will take about 147 seconds to process an image of size $1000 \times 750 \times 3$ (patch size: $7 \times 7 \times 3$) under the MATLAB

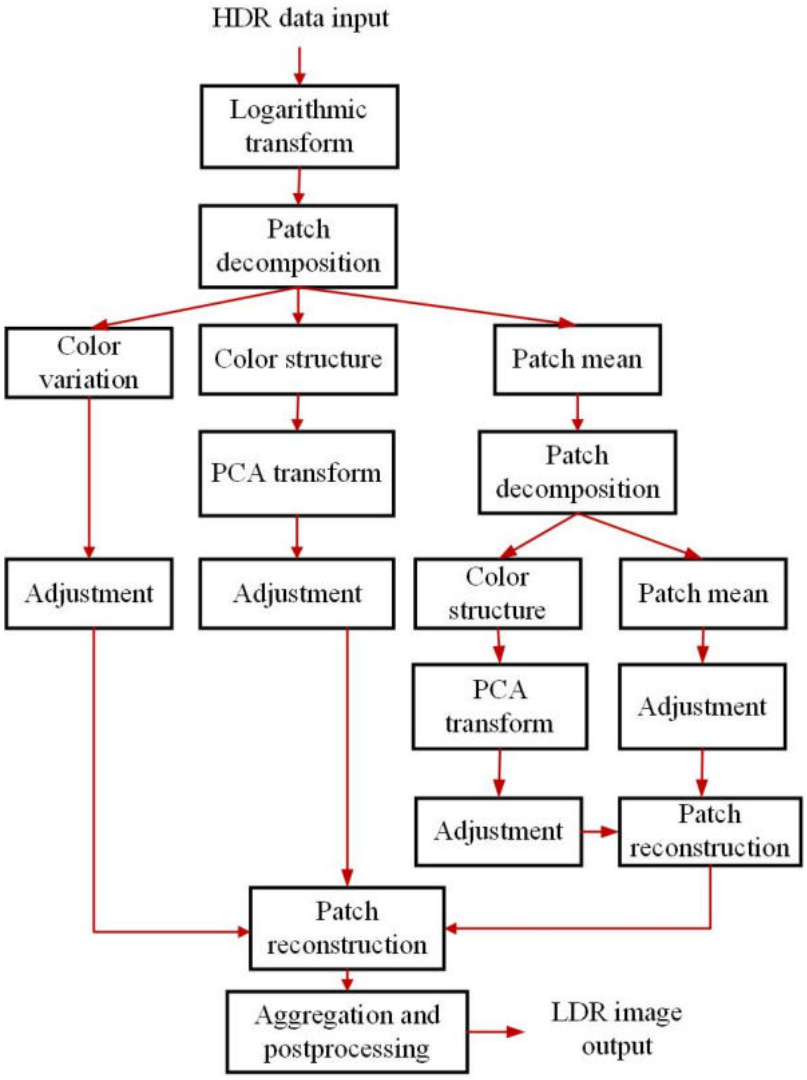


Figure 2.5: The two-scale implementation flow chart of the proposed method.

R2014a programming environment on a PC equipped with an i7-4790K CPU, 4G HZ and 32GB memory.

To reduce the computational cost, we can pre-calculate the clusters and their PCA transform matrices using an external dataset, as illustrated in Fig. 2.7. We use the Kodak database[1] as the training dataset. About 300,000 patches (patch size: $7 \times 7 \times 3$) are extracted and their color structure components are computed

[1] `http://r0k.us/graphics/kodak/`

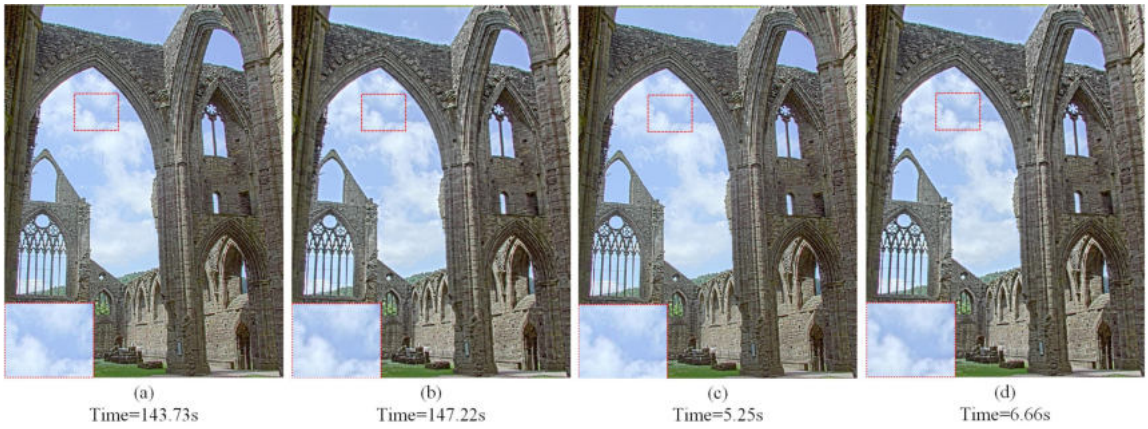


Figure 2.6: (a) and (b) are the tone mapped images by single-scale and two-scale decompositions, respectively, and (c) and (d) are the single-scale and two-scale results by off-line pre-learning of the PCA transforms.

for clustering. For each cluster, we have a cluster mean and its PCA transform matrix. In the test stage, for each patch of the input HDR image, we determine its corresponding cluster based on the minimum Euclidean distance between its color structure component and the centroids of clusters. Then the PCA transform matrix of that cluster is used to process that patch. Without the online clustering, the running time of our method is significantly improved. On average, it costs about 7 seconds to process an image of size $1000 \times 750 \times 3$, about 21 times faster than the online version of our method. In Fig. 2.6(c) and Fig. 2.6(d), we show the single-scale and two-scale tone mapping results by our offline method. We can see that the offline method achieves similar tone mapping results to the online method in terms of objective assessment (See Table 2.2 and Table 2.3).)

2.3 Experimental results and discussions

2.3.1 Implementation details

Our method is a patch based approach, and we need to fix the patch size first. Based on our experimental experience, setting the patch size from $5 \times 5 \times 3$ to $8 \times 8 \times 3$ will

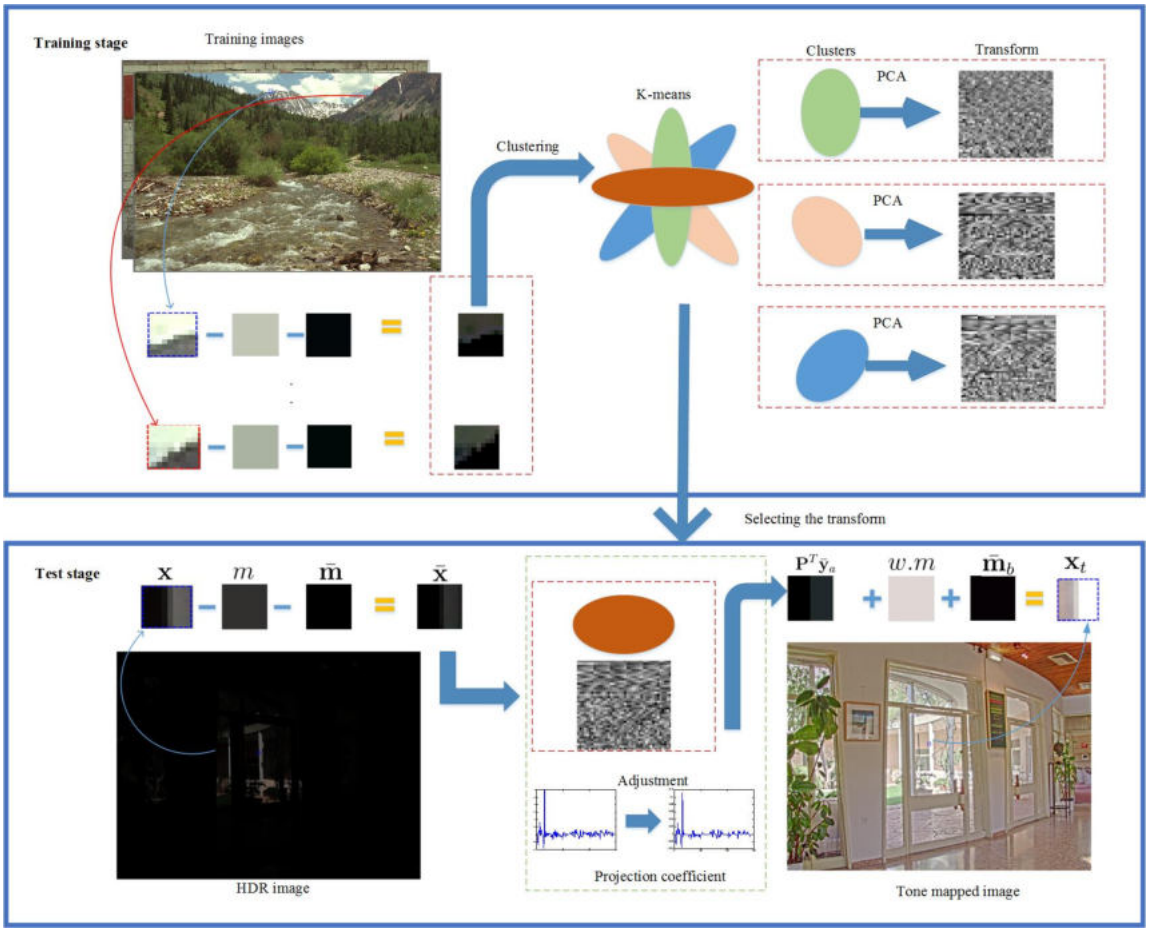


Figure 2.7: Top box: the offline patch clustering and PCA transform learning by using an external dataset. Bottom box: the online cluster selection and tone mapping.

lead to similar results, and we set the patch size to $7 \times 7 \times 3$ in all our experiments. We extract the patches from an image with stride 2 in both horizontal and vertical directions. For clustering, we use the K-means algorithm [130, 14] with initial cluster number 100 for scale 1 and 50 for scale 2. Note that some small clusters will be merged in the clustering process so that the final number of clusters will be less than 100 and 50 on the two scales. For our offline clustering method, the final numbers of clusters are 83 (scale 1) and 13 (scale 2), respectively.

The parameter a in Eq. 2.8 controls the adjustment of local structures. For simplicity, we set a the same for both the two scales. Fig. 2.8 shows the tone mapping

Figure 2.8: The impact of parameter a on the reconstruction of image local structure.

Figure 2.9: The impact of parameter b on the reconstruction of local color appearance.

results by letting a be 2, 6, 10, 20, respectively. We can see that a bigger a will make the local contrast stronger, but a too big a will make local structures and colors unnatural. We choose $a = 6$ in our experiment to achieve a good balance between contrast enhancement and color/structure preservation.

The parameter b in Eq. 2.9 controls the adjustment of local color appearance. Fig. 2.9 shows the tone mapping results by letting b be 2, 4, 8, 16, respectively. We can see that a too big b will lead to over-saturation, while a too small b will lead to under-saturation. We choose $b = 4$ in our experiments.

Finally, the parameter $w \in [0, 1]$ in Eq. 2.10 is used to adjust the luminance of the tone mapped image. Clearly, the image luminance will be lower with a smaller w. We set $w = 0.8$ based on experimental experience.

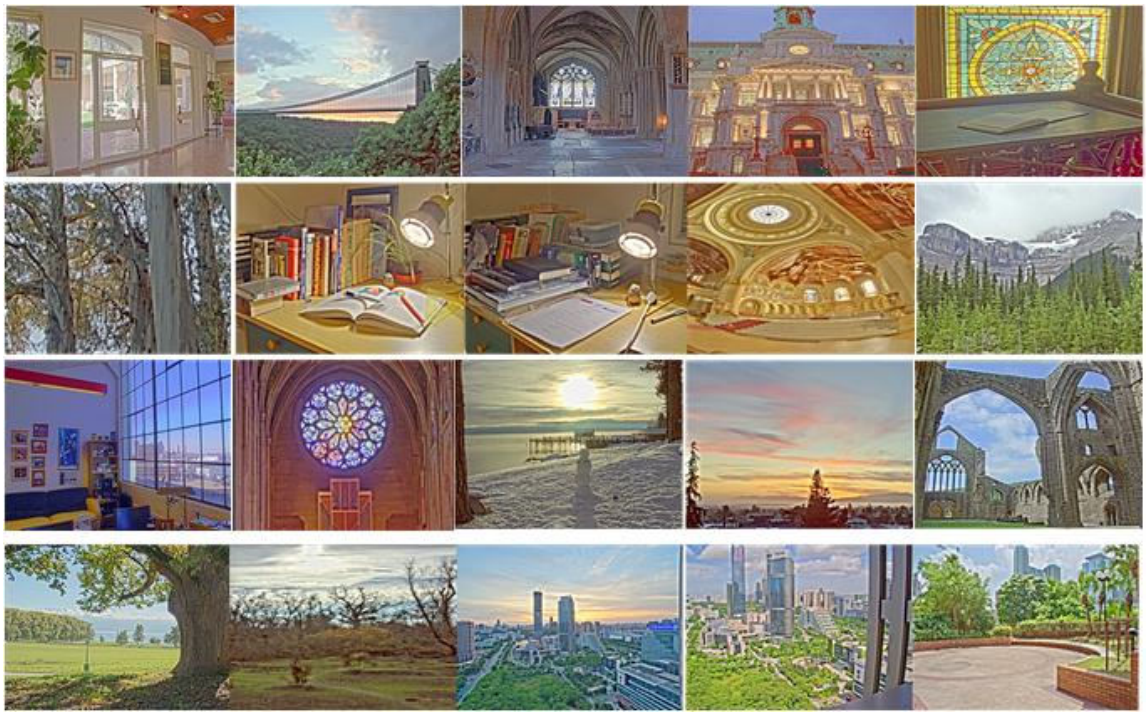


Figure 2.10: Source image scenes used in our experiment. The HDR data are represented by the tone mapped results for better visualization.

2.3.2 Test data and comparison algorithms

To verify the effectiveness of the proposed method, we collect 17 sets of widely used HDR image data from links [2, 3, 4] and capture 3 sets of HDR image data by two digital cameras (Sony a7 and DJI Phantom3). The scenes of the 20 sets of HDR images are shown in Fig. 2.10. These 20 images cover both outdoor and indoor scenes, as well as different objects such as trees, sky, sun, cloud, books, and windows.

We compare our algorithm with 7 representative tone mapping algorithms, including "Mantiuk" [80], "Drago" [15], "Fattal" [22], "Kuang" [54], "Farbman" [21], "Shan" [103], and "Shibata" [108]. The source codes of these comparison methods are publicly available in the"HDR-Toolbox" [5] or provided in the authors' home-

[2] `http://www.ok.ctrl.titech.ac.jp/res/IC/ProxPoisson/ProxPoisson.html`

[3] `http://cadik.posvete.cz/tmo/`

[4] `https://people.csail.mit.edu/sparis/publi/2011/siggraph/`

Table 2.1: Average execution time in seconds on 5 scenes of size $713 \times 535 \times 3$

Alg	Drago	Fattal	Kuang	Frabman	Shan	Shibata	Our
Env	MATLAB	MATLAB	MATLAB	MATLAB	MATLAB	MATLAB	MATLAB
Time (s)	0.13	1.15	1.23	2.89	10.28	15.01	3.86

pages[5,6,7]. We use the default parameters of those codes which were optimized by the authors. The running time of competing algorithms are summarized in Table 2.1, from which we can see that our two-scale offline method is slower than "Drago" [15], "Fattal" [22], "Kuang" [54], and "Farbman" [21], but faster than "Shan" [103], and "Shibata" [108]. Since "Mantiuk" [80] et al's method is implemented by HDR Luminance[8], we do not report it in running time comparison.

2.3.3 Objective evaluation

Since there is not a groundtruth LDR image for the HDR data, classical objective measures such as PSNR cannot be used to evaluate the quality of tone mapped images and the performance of a tone mapping algorithm. Recently, researchers have proposed some objective measures [3, 125, 87, 112, 55, 30] to evaluate the tone mapping results. The objective metrics TMQI [125] and FSITM [87] are employed in our manuscript and they are based on structural similarity (SSIM) [117] and feature similarity(FSIM) [132]. TMQI combines SSIM-motivated structural fidelity with statistical naturalness to assess the tone mapped images. FSITM measures local phase similarity of the original HDR and the tone mapped LDR image. Apart from the 7 representative methods [80, 15, 22, 54, 21, 103, 108], we also list the results of the baseline Log and Exp operators in the "HDR-toolbox" [5]. The TMQI and FSITM results are shown in Table 2.2 and Table 2.3, respectively, where Ours1, Ours2 and

[5] `http://www.cse.cuhk.edu.hk/leojia/programs/optimize_tone_mapping_code.zip`

[6] `http://www.cs.huji.ac.il/~danix/epd/`

[7] `http://www.ok.ctrl.titech.ac.jp/res/IC/ProxPoisson/ProxPoisson.html`

[8] `http://qtpfsgui.sourceforge.net/`

Table 2.2: The TMQI scores of the tone mapping images.

Source	Log	Exp	[80]	[15]	[22]	[54]	[21]	[103]	[108]	Ours1	Ours2	Ours3
1	0.550	0.866	0.889	0.907	0.701	0.823	0.918	0.843	0.866	0.934	0.939	**0.940**
2	0.698	0.748	0.867	0.842	0.844	**0.928**	0.727	0.805	0.908	0.871	0.872	0.876
3	0.576	0.886	0.936	0.896	0.818	0.855	0.**974**	0.973	0.850	0.933	0.939	0.938
4	0.630	0.810	0.912	0.951	0.790	**0.911**	0.844	0.835	0.774	0.857	0.856	0.869
5	0.709	0.922	0.699	**0.960**	0.808	0.817	0.844	0.905	0.781	0.944	0.934	0.932
6	0.807	0.813	0.852	**0.958**	0.885	0.880	0.821	0.933	0.771	0.809	0.811	0.816
7	0.759	0.910	0.916	0.921	0.758	0.843	0.844	0.936	0.822	**0.959**	0.954	0.952
8	0.740	0.895	0.874	0.870	0.752	0.824	0.807	0.919	0.813	0.945	**0.949**	0.946
9	0.639	0.856	0.896	**0.948**	0.769	0.832	0.807	0.921	0.755	0.909	0.898	0.906
10	0.729	0.887	0.890	0.870	0.880	**0.962**	0.719	0.960	0.930	0.890	0.884	0.885
11	0.706	0.882	0.923	0.952	0.854	**0.952**	0.882	0.804	0.838	0.943	0.933	0.916
12	0.536	**0.938**	0.888	0.938	0.748	0.796	0.776	0.869	0.781	0.892	0.887	0.885
13	0.534	0.854	0.915	0.944	0.786	0.790	0.928	0.950	0.839	0.952	**0.954**	0.953
14	0.828	0.782	**0.878**	0.788	0.803	0.867	0.521	0.789	0.844	0.812	0.817	0.817
15	0.773	0.913	0.908	0.901	0.903	**0.986**	0.741	0.886	0.890	0.909	0.899	0.898
16	0.777	0.868	0.949	0.933	**0.953**	0.946	0.774	0.905	0.823	0.930	0.828	0.835
17	0.667	0.879	0.883	0.906	0.912	**0.979**	0.718	0.957	0.908	0.953	0.947	0.945
18	0.773	0.817	0.921	0.873	0.804	**0.957**	0.662	0.897	0.922	0.919	0.919	0.915
19	0.800	0.862	**0.960**	0.881	0.936	0.939	0.728	0.909	0.851	0.846	0.845	0.845
20	0.790	0.880	**0.976**	0.916	0.941	0.952	0.751	0.920	0.859	0.848	0.847	0.847
Average	0.701	0.863	0.897	**0.908**	0.832	0.892	0.790	0.896	0.841	0.903	0.896	0.896

Ours3 represent the single-scale, two-scale, and the off-line two-scale implementations of our method respectively. For each image, the best result is highlighted in bold face.

2.3.4 Subjective comparison

Let's then present some visual comparisons of the competing methods. For our method, we present the results by the offline two-scale implementation. Figs. 2.11-2.14 show the tone mapped images of scenes 7, 9, 17, and 18 (see Fig. 2.10), respectively.

The results by "Mantiuk" [80] present the loss of details especially in dark regions. For example, in the close-up images in Fig. 2.11 and Fig. 2.14, the books and trees cannot be seen. The adaptive global method "Drago" [15] presents better results, but it suffers from the loss of local contrast. One can see from Fig. 2.13 that the contrast of tree branches and cloud background is low. Fattal et al's method [22] has

Table 2.3: The FSITM scores of the tone mapping images.

Source	Log	Exp	[80]	[15]	[22]	[54]	[21]	[103]	[108]	Ours1	Ours2	Ours3
1	**0.863**	0.783	0.852	0.829	0.821	0.857	0.848	0.774	0.779	0.802	0.804	0.803
2	0.756	0.779	0.849	0.854	0.736	0.868	0.771	0.809	0.844	0.852	0.853	**0.855**
3	**0.927**	0.857	0.903	0.896	0.861	0.897	0.894	0.868	0.823	0.830	0.839	0.838
4	0.696	0.846	0.901	**0.915**	0.732	0.917	0.901	0.842	0.869	0.901	0.901	0.902
5	0.795	0.726	0.719	0.790	0.748	**0.823**	0.801	0.715	0.759	0.792	0.791	0.792
6	0.922	0.879	0.932	**0.951**	0.783	0.948	0.916	0.856	0.905	0.930	0.931	0.932
7	0.811	0.808	0.866	0.869	0.723	0.878	0.871	0.831	0.826	0.872	0.872	**0.873**
8	0.711	0.802	0.855	0.860	0.717	**0.872**	0.861	0.832	0.826	0.864	0.864	0.864
9	0.803	0.827	0.913	**0.923**	0.757	0.932	0.910	0.877	0.884	0.920	0.920	0.921
10	0.863	0.868	0.904	**0.924**	0.760	0.924	0.797	0.895	0.896	0.906	0.907	0.908
11	0.735	0.757	0.838	0.846	0.724	0.831	0.835	0.732	0.807	0.842	0.846	**0.852**
12	0.606	0.803	0.861	0.875	0.750	**0.888**	0.857	0.838	0.853	0.878	0.880	0.881
13	0.797	0.748	0.819	0.800	0.811	**0.837**	0.821	0.774	0.780	0.785	0.788	0.788
14	0.818	0.778	0.834	0.838	0.743	**0.861**	0.602	0.792	0.845	0.840	0.840	0.846
15	**0.906**	0.817	0.884	0.862	0.799	0.872	0.777	0.753	0.837	0.833	0.835	0.835
16	0.803	0.802	0.914	0.926	0.762	**0.930**	0.872	0.835	0.882	0.921	0.921	0.927
17	0.796	0.835	0.907	0.914	0.749	**0.926**	0.804	0.890	0.898	0.908	0.909	0.909
18	0.825	0.795	0.843	0.852	0.747	0.874	0.722	0.821	0.827	0.855	0.858	**0.861**
19	0.864	0.856	0.874	**0.902**	0.755	0.889	0.785	0.837	0.866	0.882	0.886	0.889
20	0.870	0.821	**0.915**	0.913	0.764	0.903	0.810	0.835	0.887	0.906	0.909	0.909
Average	0.808	0.809	0.869	0.877	0.762	**0.886**	0.823	0.820	0.845	0.866	0.868	0.870

Figure 2.11: The tone mapping results on image 7 (refer to Fig. 2.10) by competing tone mapping operators. From (a) to (h): results by "Mantiuk" [80], "Drago" [15], "Fattal" [22], "Kuang" [54], "Farbman" [21], "Shan" [103], "Shibata" [108], and ours. From (i) to (p): the close-ups of (a)-(h).

Figure 2.12: The tone mapping results on image 9 (refer to Fig. 2.10) by competing tone mapping operators. From (a) to (h): results by "Mantiuk" [80], "Drago" [15], "Fattal" [22], "Kuang" [54], "Farbman" [21], "Shan" [103], "Shibata" [108], and ours.

the problem of detail and contrast loss such as the wall in Fig. 2.11 and green tree in Fig. 2.14. Kuang et al's method [54] shows much distortion of color appearance, although it preserves well local details and contrasts. For instance, it produces a purple color of sky in Fig. 2.13, which is not natural. The tone mapped images by multi-scale decomposition based method "Farbman" [21] suffer from information loss in some regions, such as the sky in Fig. 2.13 and Fig. 2.14. Shan et al's method [103] over-smooths much the image local textures. There are neither clear contours of the cloud in Fig. 2.13 nor fine structures of tree leaves in Fig. 2.14. Shibata et al's method [108] shows good local contrast but meanwhile generates much visual

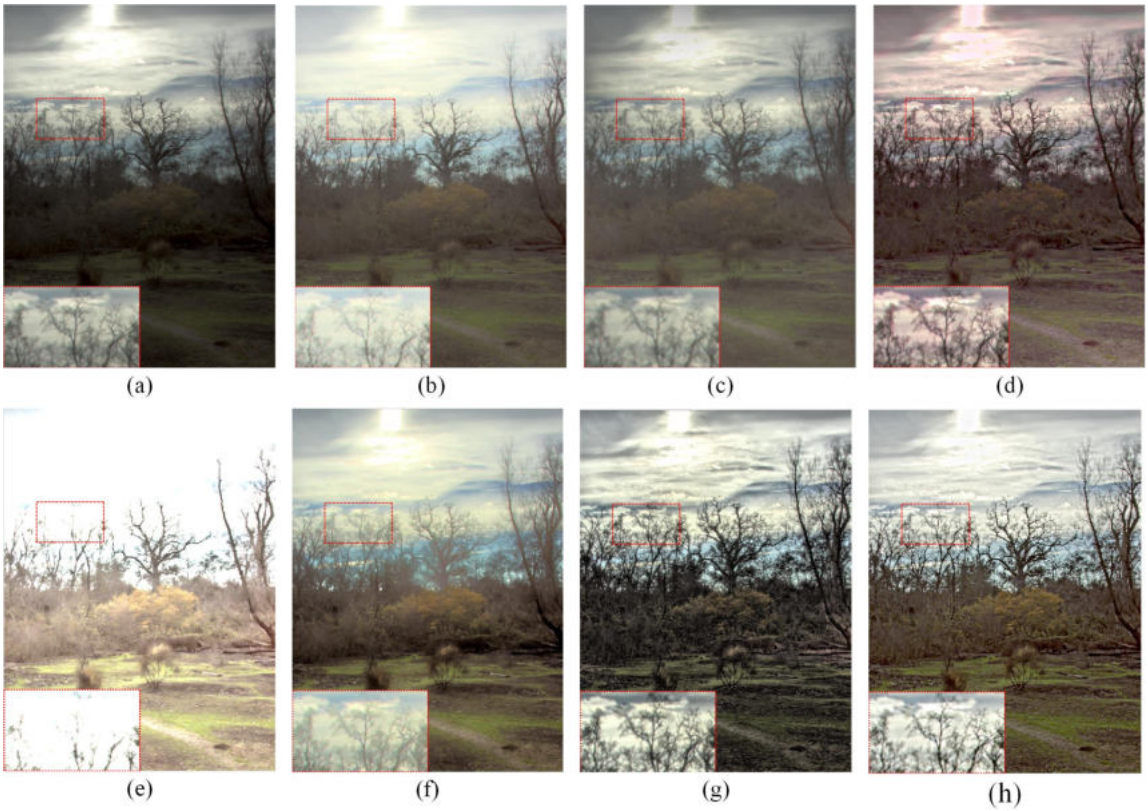


Figure 2.13: The tone mapping results on image 17 (refer to Fig. 2.10) by competing tone mapping operators. From (a) to (h): results by "Mantiuk" [80], "Drago" [15], "Fattal" [22], "Kuang" [54], "Farbman" [21], "Shan" [103], "Shibata" [108], and ours.

artifacts. The surfaces of the wall and desk in Fig. 2.11 and the roofs in Fig. 2.12 are over-exaggerated.

Compared with the above methods, our method demonstrates competitive visual quality with good local structure preservation and color reproduction. For instance, in Fig. 2.11 the local details and contrast labelled in the red box can be seen clearly with decent overall visual effect. Furthermore, the colors of trees, cloud and grass look natural and saturated. This is mainly because our method clusters image patches based on their local colors and structures and it processes each patch adaptively based on the color and structure statistical information in that cluster.

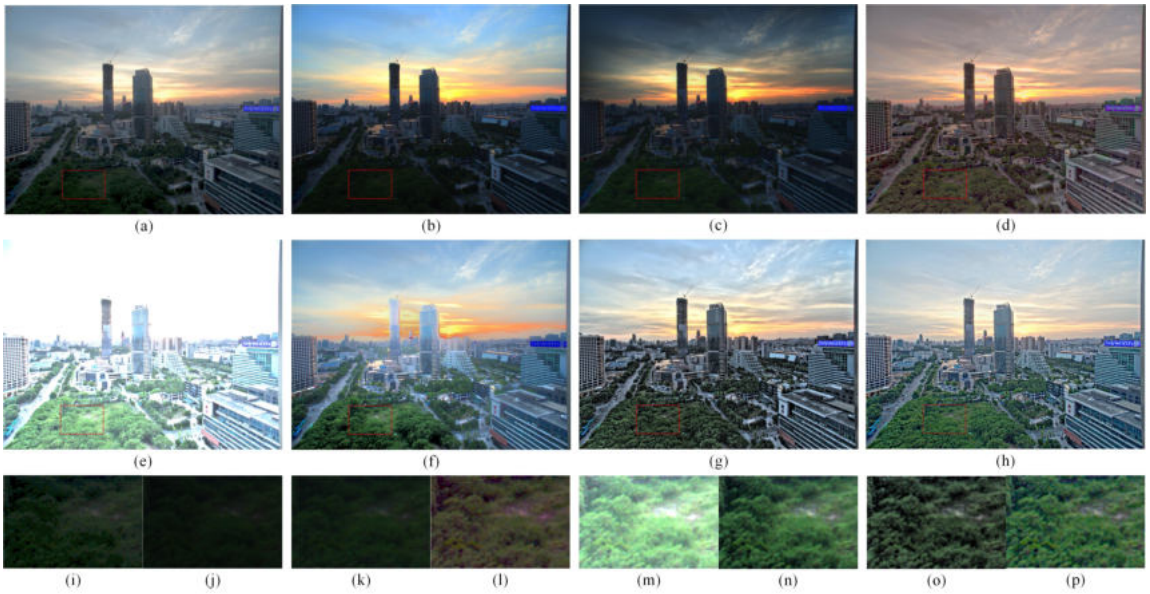


Figure 2.14: The tone mapping results on image 18 (refer to Fig. 2.10) by competing tone mapping operators. From (a) to (h): results by "Mantiuk" [80], "Drago" [15], "Fattal" [22], "Kuang" [54], "Farbman" [21], "Shan" [103], "Shibata" [108], and ours. From (i) to (p): the close-ups of (a)-(h).

2.3.5 Subjective Study

A formal subjective study is conducted to further evaluate the proposed tone mapper and compared methods. The subjective testing was operated in an indoor environment with stable illumination as shown in Fig. 2.15. We adopted the strategy in [79] in our subjective testing. The tone mapped images of 20 scenes by 8 representative algorithms are shown on a PA328 Display, 32-inch (7680*4320), controlled by a Mac Pro with Intel Core i5 2.9GHz CPU. A total number of 17 volunteer subjects, including 8 females and 9 males, were asked to give an integer score ranging from 1-10 to each image shown on the display, where 1 means the worst visual quality and 10 means the best visual quality.

The mean and std of mean opinion score (MOS) values are shown in Fig. 2.16. It can be seen that our method and Shibata et al's method have much better performance than other competing methods. The MOS of our method is 7.50 with std

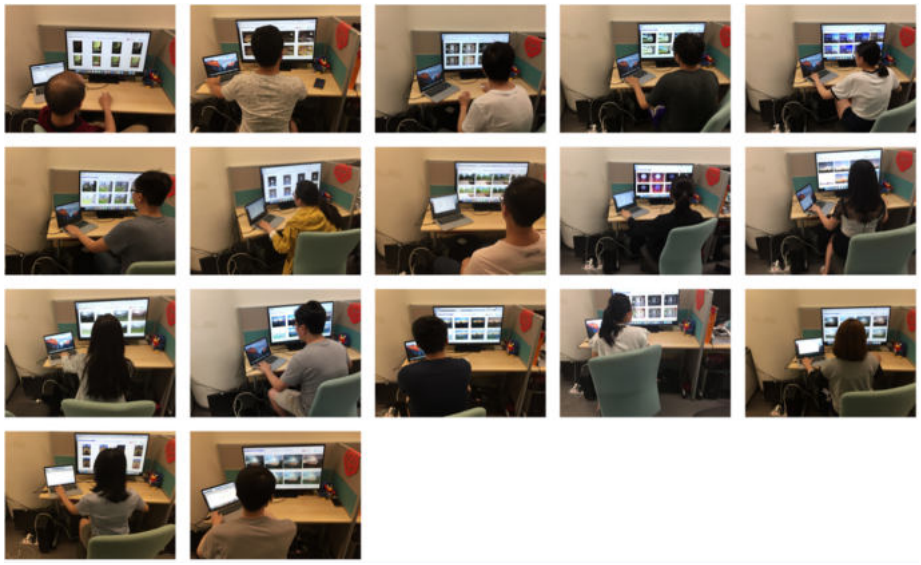


Figure 2.15: The environment and 17 subjects participated in the subjective experiments.

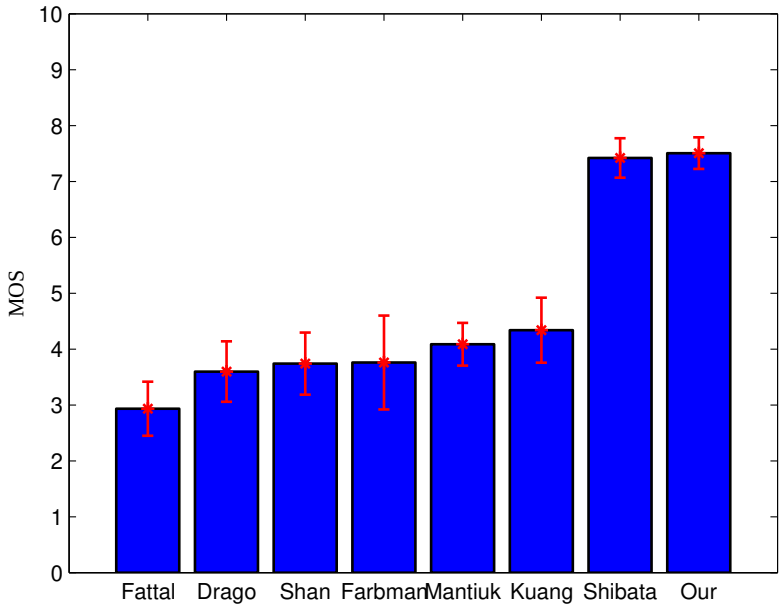


Figure 2.16: Mean and std of subjective rankings of the 8 competing tone mapping algorithms.

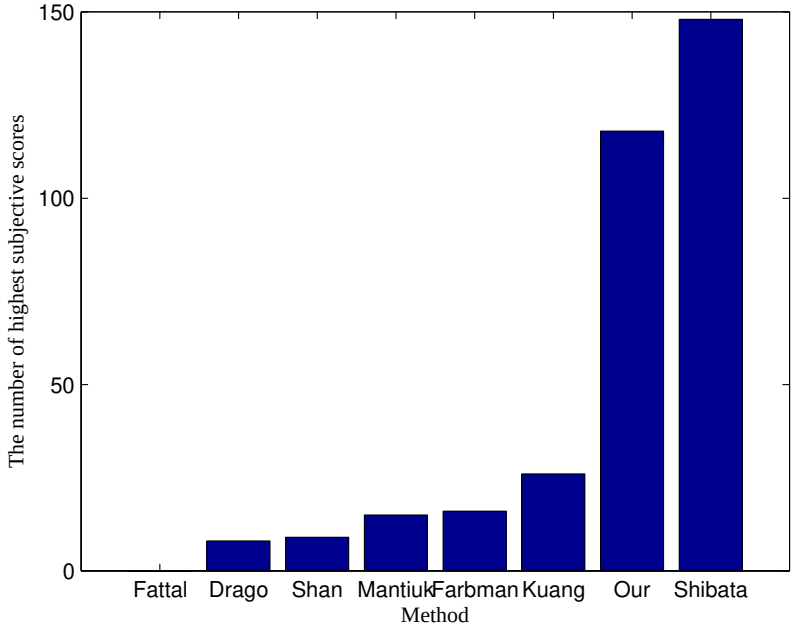


Figure 2.17: The number of highest subjective scores obtained by different methods.

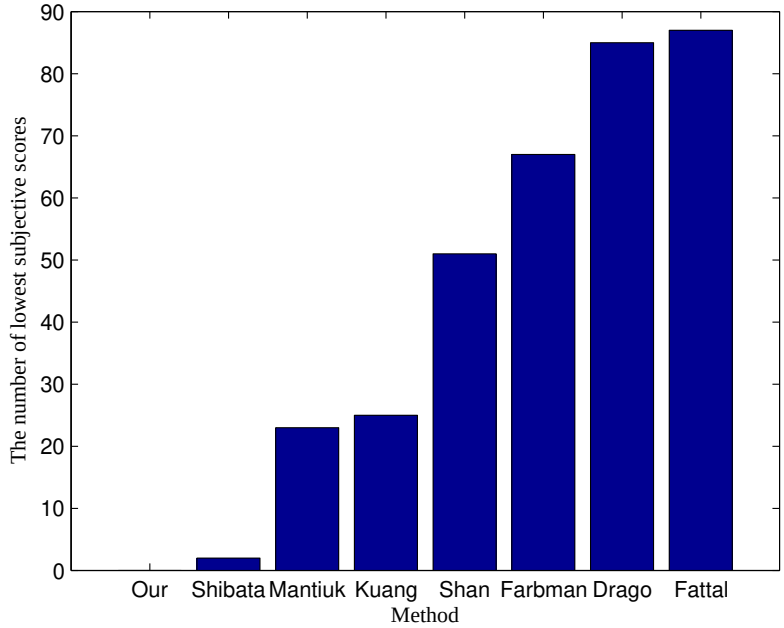


Figure 2.18: The number of lowest subjective scores obtained by different methods.

0.56, while that of Shibata et al's method is 7.42 with std 0.71. In the subjective experiments, our method obtains 118 highest subjective scores and 0 lowest subjective score among 340 highest and lowest scores. The distributions of numbers of highest and lowest scores by different methods are shown in Fig. 2.17 and Fig. 2.18. Overall, our method demonstrates highly competitive and stable tone mapping performance.

It should be pointed out that the subjective testing results are not well consistent with the objective metrics used in this chapter. Existing objective metrics for tone mapping operators are primarily focused on structural similarity [125], feature similarity [87], visibility [3, 112, 55], contrast [112, 55], naturalness [125, 87], and chrominance [112]. These quality measures are derived from general image quality assessment methods and they may not be suitable for the tone mapping problem. It is still a challenging issue to design a faithful perceptual quality measure to assess tone mapping operators. In addition, we found that the naturalness index should not be over-emphasized for evaluating tone mapping methods via our subjective experiments, and that the color information plays an important role in assessing tone mapped images.

2.4 Conclusion

In this chapter, we presented a clustering based content and color adaptive tone mapping method. Different from previous methods which are mostly filtering based, our method works on image patches, and it decomposes each patch into three components: patch mean, color variation and color structure. Based on the color structure component, we clustered image patches into clusters, and calculated the PCA transform matrix for each cluster. The patches were then transformed into its PCA domain, and the s-shaped arctan function was used to adjust their PCA coefficients. We further extended our method to two scales and proposed an offline clustering

implementation to improve its fine-texture preservation and efficiency. Experiments on 20 sets of HDR data demonstrated the superior performance of our method to representative tone mapping methods.

Chapter 3

Multi-Scale Fast Structural Patch Decomposition for Multi-Exposure Image Fusion

Exposure bracketing is crucial to high dynamic range imaging, but it is prone to halos for static scenes and ghosting artifacts for dynamic scenes. The recently proposed structural patch decomposition for multi-exposure fusion (SPD-MEF) has achieved superior performance in deghosting. However, it is computationally expensive and suffers from visible halo artifacts, while its relationship to other MEF methods is unclear. Here we show that an unnormalized version of SPD-MEF is closely related to standard pixel-level MEF methods as well as the classical two-layer decomposition method for MEF. Moreover, it avoids explicitly performing structural patch decomposition, which achieves an order of 30× speed-up. We further develop a multi-scale fast SPD-MEF method, which effectively reduces the halo artifacts. Experimental results demonstrate the effectiveness of our multi-scale fast SPD-MEF in terms of speed and quality.

Figure 3.1: Left column: Mertens09 [82]. Middle column: SPD-MEF [75]. Right column: Our method. One can see that our method can suppress ghost artifacts and halo artifacts better than Mertens09 and SPD-MEF.

3.1 Introduction

Recently, Ma *et al.* proposed the structural patch decomposition for MEF (SPD-MEF) [75] that demonstrates reliable deghosting performance over a wide range of dynamic scenes. The consistent improvement of SPD-MEF in visual quality has been verified by MEF-SSIM [78], a widely used objective quality metric for MEF, and in two independent subjective experiments [7, 20]. Although faster than many HDR deghosting algorithms, SPD-MEF still takes seconds (even minutes) to fuse high-resolution sequences, and therefore is not suitable for real-time mobile applications. In addition, it produces visible halo artifacts for some natural scenes, where the difference in dynamic range between the foreground and the background is large (see Fig. 3.1).

A predominant problem of MEF is the introduction of the ghosting artifacts when dealing with dynamic scenes that contain moving objects (see Fig 3.1). While

many MEF algorithms (also referred to as HDR degosting methods) are able to produce ghost-free images, they come with their own disadvantages such as substantial computational complexity due to the need of solving a global optimization problem [39, 60, 90], or suboptimal visual quality due to excessive reliance on the reference exposure for inconsistent motion rejection [102, 70, 95]. Recently, Ma *et al.* described the structural patch decomposition for MEF (SPD-MEF) [75] that demonstrates reliable deghosting performance over a wide range of dynamic scenes. The consistent improvement of SPD-MEF in visual quality has been verified by MEF-SSIM [78], a widely used objective quality metric for MEF, and in two independent subjective experiments [7, 20]. Although faster than most HDR deghosting algorithms, SPD-MEF still takes seconds (even minutes) to fuse high-resolution sequences, and therefore is not suitable for real-time mobile applications. In addition, it produces visible halo artifacts for some natural scenes, where the difference in dynamic range between the foreground and the background is large (see Fig. 3.1).

In this chapter, we study SPD-MEF [75] to gain a better understanding of its behavior. Our empirical analysis shows that we can skip the normalization step when fusing signal structures without introducing noticeable differences to the original scheme. By further incorporating the signal strength into the weight, we avoid explicitly performing structural patch decomposition, leading to an acceleration scheme that runs about 30 times faster. We rewrite patch aggregation as mean filtering of the weight map at each exposure, and arrive at a formulation that is closely related to standard pixel-level MEF methods [82, 106, 69, 27, 89, 53, 1] and the two-layer decomposition for MEF [96, 65]. The main difference lies in how their weights are designed and computed. Finally, we propose the multi-scale fast SPD-MEF approach by progressively downsamping and processing the mean intensity images, which effectively reduces the halo artifacts with little additional computation. Experiments on a wide range of static and dynamic scenes show that our multi-scale fast SPD-MEF

algorithm consistently produces HDR images with little ghosting and halo artifacts while being the fastest among state-of-the-arts.

In this section, we first revisit the algorithm flow of SPD-MEF [75], and then show that an unnormalized approximation permits a neat acceleration scheme, whose relationship to other approaches is also much clearer. We then develop a multi-scale fast SPD-MEF approach with reduced halo artifacts.

3.2 SPD-MEF

Let's briefly describe how SPD-MEF [75] computes the fused image. The core idea of SPD-MEF for static scenes is to decompose an image patch of dimension N into three conceptually independent components: mean intensity, signal strength, and signal structure

$$\begin{aligned}\mathbf{x} &= l \cdot \mathbf{1} + \|\mathbf{x} - l\| \cdot \frac{\mathbf{x} - l}{\|\mathbf{x} - l\|} \\ &= l \cdot \mathbf{1} + \|\bar{\mathbf{x}}\| \cdot \frac{\bar{\mathbf{x}}}{\|\bar{\mathbf{x}}\|} \\ &= l \cdot \mathbf{1} + c \cdot \mathbf{s}, \end{aligned} \tag{3.1}$$

where $\mathbf{1}$ is an N-dimensional vector of all ones and $\|\bar{\mathbf{x}}\|$ denotes the l_2-norm of the mean-removed patch $\bar{\mathbf{x}}$. l and c are two scalars representing mean intensity and signal strength, respectively. $\mathbf{s}$ is a unit-length vector, whose direction encodes signal structure. The desired patch of the output fused image can be obtained by determining the three components separately and inverting the decomposition. Specifically, assuming the input sequence has K exposures, the desired local mean intensity is computed by

$$\hat{l} = \sum_{k=1}^{K} \alpha_k l_k, \tag{3.2}$$

where the weight $\alpha_k \geq 0$ depends on the local and global mean intensities of the k-th exposure, and $\sum_k \alpha_k = 1$. The desired local signal strength is computed as the largest one across exposures

$$\hat{c} = \max_{1 \leq k \leq K} \|\bar{\mathbf{x}}_k\| = \max_{1 \leq k \leq K} c_k. \tag{3.3}$$

The desired local signal structure is determined by

$$\hat{\mathbf{s}} = \frac{\bar{\mathbf{s}}}{\|\bar{\mathbf{s}}\|}, \quad \text{where} \quad \bar{\mathbf{s}} = \sum_{k=1}^{K} \beta_k \mathbf{s}_k. \tag{3.4}$$

The weight β_k is proportional to $\|\bar{\mathbf{x}}_k\|$, and $\sum_k \beta_k = 1$. After this, we are able to compute the desired local patch

$$\hat{\mathbf{x}} = \hat{l} \cdot \mathbf{1} + \hat{c} \cdot \hat{\mathbf{s}}. \tag{3.5}$$

SPD-MEF performs patch aggregation by simply averaging all overlapping pixel values to obtain the final fused image [75].

3.3 Fast SPD-MEF

Most computational cost of SPD-MEF comes from the structural patch decomposition in Eq. (3.1), which has a complexity of $\mathcal{O}(NMK)$, where N is the patch size, M is the number of pixels in each exposure, and K is the number of multi-exposure images. In this section, we will show that the complexity can be reduced to $\mathcal{O}(MK)$.

We first analyze $\bar{\mathbf{s}}$, which is a convex combination of K unit length vectors. The norm of $\bar{\mathbf{s}}$ satisfies

$$\|\bar{\mathbf{s}}\| = \left\| \sum_{k=1}^{K} \beta_k \mathbf{s}_k \right\| \leq \sum_{k=1}^{K} \beta_k \|\mathbf{s}_k\| = 1, \tag{3.6}$$

which can be easily proved by induction using triangular inequality and absolute homogeneity of the norm. The equality holds for arbitrarily chosen $\{\beta_k\}$ when all

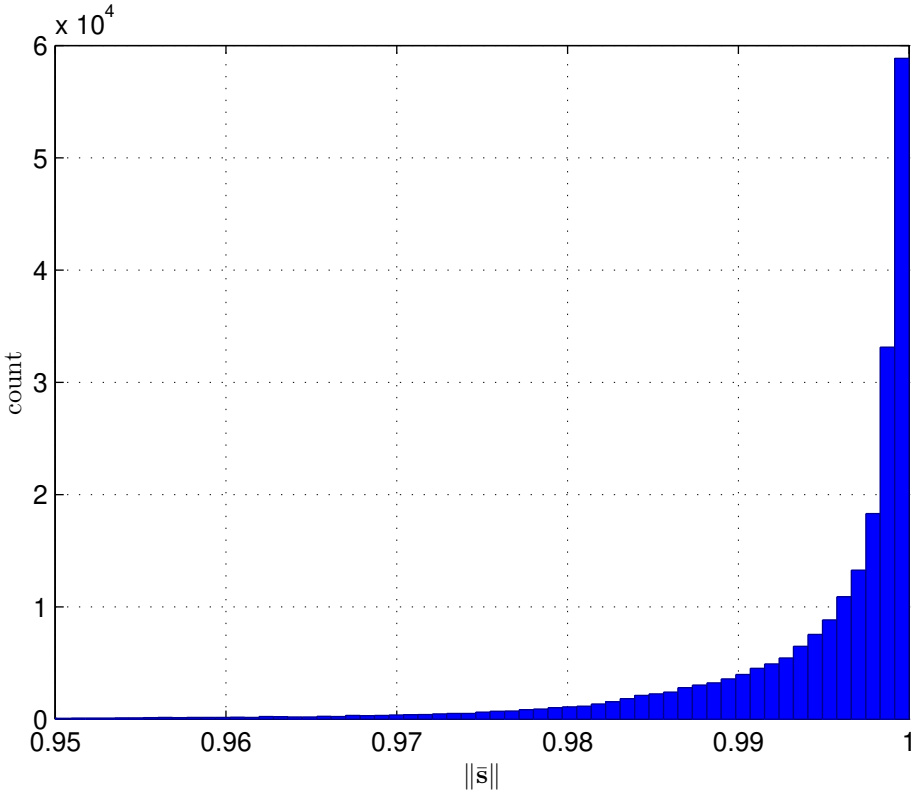


Figure 3.2: The histogram of $\|\bar{\mathbf{s}}\|$ computed from six static scenes.

signal structures are identical. If some $\mathbf{s}_k$ points to a different direction, we may still achieve the equality by assigning the corresponding β_k to zero. Empirically, we find that $\|\bar{\mathbf{s}}\|$ computed by SPD-MEF is close to one for all co-located patches from different sequences (see the histogram in Fig. 3.2). This is expected because as long as the set of $\{\mathbf{x}_k\}$ are not under-/over-exposed, the corresponding exposure-invariant $\{\mathbf{s}_k\}$ have the same structure, leading to $\|\bar{\mathbf{s}}\| \approx 1$. For under-exposed regions, $\mathbf{s}_k$ mainly contains amplified noise structure; for over-exposed regions, $\mathbf{s}_k$ is nearly flat, *i.e.*, $\frac{1}{\sqrt{N}}\mathbf{1}$. In either case, $\mathbf{s}_k$ points to a different direction from the true signal structure. Fortunately, the corresponding β_k computed by SPD-MEF is close to zero, giving rise to $\|\bar{\mathbf{s}}\| \approx 1$. This implies that normalizing the desired signal structure has little impact to the overall computation, and SPD-MEF without the normalization step would deliver essentially the same visual results (see Fig. 3.3).

(a)

(b) (c)

Figure 3.3: SPD-MEF with and without normalization. (a) Image sequence "Landscape" (courtesy of HDRsoft). (b) With normalization. (c) Without normalization. The visual similarity between the two images is verified by an SSIM [117] value of 0.999.

We proceed by substituting (3.2), (3.4) into (3.5)

$$\begin{aligned}\hat{\mathbf{x}} &\approx \sum_{k=1}^{K} \left(\alpha_k l_k \cdot \mathbf{1} + \hat{c}\beta_k \cdot \mathbf{s}_k\right) \\ &= \sum_{k=1}^{K} \left(\alpha_k l_k \cdot \mathbf{1} + \frac{\hat{c}\beta_k}{\|\bar{\mathbf{x}}_k\|} \cdot \bar{\mathbf{x}}_k\right) \\ &= \sum_{k=1}^{K} \big(\alpha_k l_k \cdot \mathbf{1} + \gamma_k \cdot (\mathbf{x}_k - l_k)\big), \end{aligned} \tag{3.7}$$

where $\gamma_k = \frac{\hat{c}\beta_k}{\|\bar{\mathbf{x}}_k\|}$. Note that by approximating $\hat{\mathbf{s}}$ with $\bar{\mathbf{s}}$ and incorporating $\|\bar{\mathbf{x}}_k\|$ into γ_k, we avoid explicitly performing structural patch decomposition. The final image

$\hat{\mathbf{X}}$ can then be computed by

$$\hat{\mathbf{X}} = \sum_{k=1}^{K} \big(f\left(\boldsymbol{\alpha}_k \odot \mathbf{L}_k\right) + f(\boldsymbol{\gamma}_k) \odot \mathbf{X}_k - f\left(\boldsymbol{\gamma}_k \odot \mathbf{L}_k\right) \big), \tag{3.8}$$

where $f(\cdot)$ is a mean filter of dimension N and $\odot$ denotes Hadamard product. That is, instead of averaging all overlapping patch values, we equivalently smooth the weight maps with f. The mean filtering process can be implemented in linear time via box filter [36]. Consequentially, the computational complexity of SPD-MEF is reduced from $\mathcal{O}(NMK)$ to $\mathcal{O}(MK)$, independent of patch size N.

We now take a closer look at Eq. (3.8). Choosing $\boldsymbol{\alpha}_k = \boldsymbol{\gamma}_k$ yields the classic form of pixel-level MEF, with a smoothed weight map $f(\boldsymbol{\gamma}_k)$. If each pixel computes a separate mean intensity from the patch centered at it, Eq. (3.8) becomes

$$\hat{\mathbf{X}} = \sum_{k=1}^{K} \Big(f(\boldsymbol{\alpha}_k) \odot f\left(\mathbf{X}_k\right) + f(\boldsymbol{\gamma}_k) \odot \big(\mathbf{X}_k - f\left(\mathbf{X}_k\right) \big) \Big), \tag{3.9}$$

which is essentially the two-layer decomposition of images for MEF. The weight maps for the base layer and the detail layer are $f(\boldsymbol{\alpha}_k)$ and $f(\boldsymbol{\gamma}_k)$, respectively. In the original development of SPD-MEF [75], the authors speed up the algorithm by sampling patches with a stride larger than one, which can also be incorporated into Eq. (3.8)

$$\begin{aligned} \hat{\mathbf{X}} = \sum_{k=1}^{K} \big(& f\left(\boldsymbol{\alpha}_k \odot \mathbf{M}_k \odot \mathbf{L}_k\right) + f(\boldsymbol{\gamma}_k \odot \mathbf{M}_k) \odot \mathbf{X}_k \\ & - f\left(\boldsymbol{\gamma}_k \odot \mathbf{M}_k \odot \mathbf{L}_k\right) \big), \end{aligned} \tag{3.10}$$

where $\mathbf{M}_k$ is a binary mask with ones indicating patches that have been sampled.

3.4 Multi-scale fast SPD-MEF

The kernel size of the mean filter f, or equivalently the patch size, has a significant impact on the fusion performance. A small kernel usually recovers more details, but

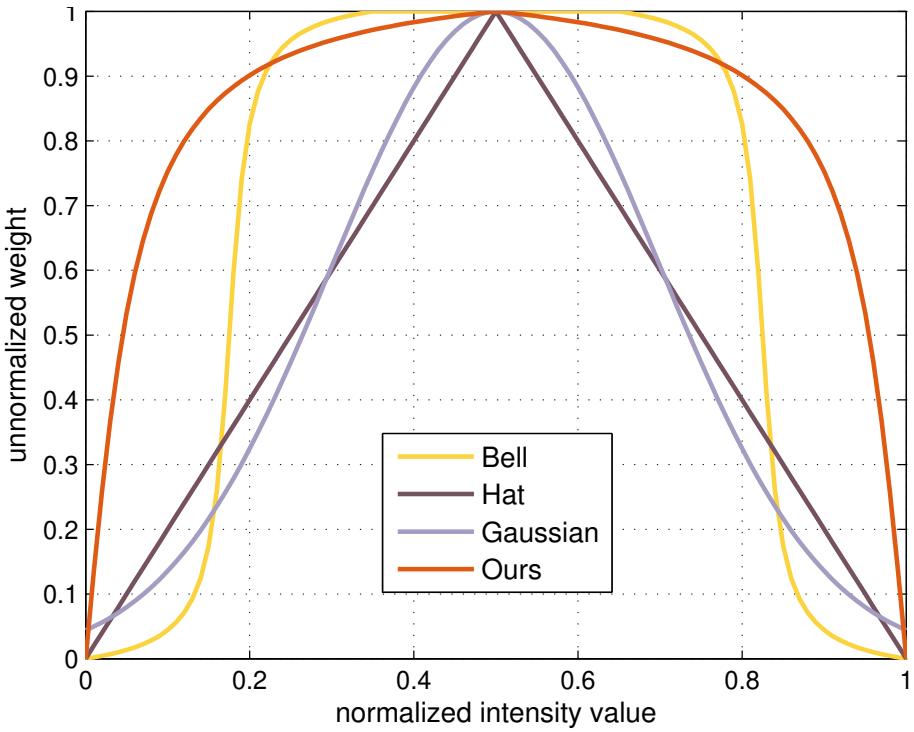


Figure 3.4: Comparison of different well-exposedness weight functions.

tends to produce noisy weight maps, resulting in spatial inconsistency of the fused image. A large kernel effectively resolves this problem at the cost of halo artifacts near strong edges due to unwanted smoothing [36].

Here we describe a multi-scale fast SPD-MEF approach to reduce halos while preserving the details at different scales. We index the original sequence as Scale 1. In Eq. (3.8), we notice that a desired detail layer that contains rich high-frequency information is computed as

$$\hat{\mathbf{D}}^{(1)} = \sum_{k=1}^{K} \left(f\left(\boldsymbol{\gamma}_k^{(1)}\right) \odot \mathbf{X}_k^{(1)} - f\left(\boldsymbol{\gamma}_k^{(1)} \odot \mathbf{L}_k^{(1)}\right) \right). \tag{3.11}$$

To make SPD-MEF multi-scale, we do not fuse $\{\mathbf{L}_k^{(1)}\}$, but to downsample them by a factor of two, from which $\hat{\mathbf{D}}^{(2)}$ that contains the fine details at Scale 2 is computed. The process is then applied recursively to the downsampled $\{\mathbf{L}_k^{(j)}\}$ until the coarsest

(a) (b)

Figure 3.5: Visual demonstration of the proposed multi-scale SPD-MEF approach on the image sequence "Arno" (courtesy of Bartlomiej Okonek). (a) Desired base layer and desired detail layers at four scales. (b) Final fused image.

scale

$$J = \lfloor \log_2 \min(H, W) \rfloor - 3 \tag{3.12}$$

is reached, where H and W represent the height and width of the sequence, respectively. The constant three is subtracted to ensure that the resolution at the coarsest scale is not too small.

We obtain the desired base layer $\hat{\mathbf{L}}^{(J)}$ at the coarsest scale by fusing $\{\mathbf{L}_k^{(J)}\}$ according to their well-exposedness. Instead of adopting Gaussian curves [82, 75], we propose a modified arctan$(\cdot)$ function as the well-exposedness measure

$$\boldsymbol{\alpha}_k^{(J)} = \frac{\arctan\left(0.5\lambda - \left|0.5 - \mathbf{L}_k^{(J)}\right| \lambda\right)}{\sum_{k=1}^{K} \arctan\left(0.5\lambda - \left|0.5 - \mathbf{L}_k^{(J)}\right| \lambda\right)}, \tag{3.13}$$

where λ is a fixed parameter. To see the difference, we compare four well-exposedness weight functions in Fig. 3.4, where we observe that our measure gives less penalty to slightly under-/over-exposed intensities. This gives us an opportunity to better preserve global brightness. The weight maps $\{\boldsymbol{\gamma}^{(j)}\}$ for $\{\hat{\mathbf{D}}^{(j)}\}$ are the same as the original SPD-MEF [75], but they are computed at their respective scales. In generating $\hat{\mathbf{D}}^{(1)}$, we compute the statistics on the RGB images by stacking the three channels together. In other words, $\hat{\mathbf{D}}^{(1)}$ contains not only the finest details but also

rich color information of the sequence, which is beneficial for creating a vivid color appearance of the fused image [75]. $\hat{\mathbf{D}}^{(j)}$ is computed from grayscale $\{\mathbf{L}_k^{(j)}\}$ and is responsible for recovering monochromatic high-frequency information at Scale j.

Finally, the fused image is obtained by progressively upsampling and adding back the desired detail layers to the base layer. Fig. 3.5 shows the intermediate results of our method at four scales along with the final output. As can be seen, our method produces natural appearance with faithful detail and color reproduction.

3.5 Handling dynamic scenes

When dealing with dynamic scenes that contains noticeable object motion, SPD-MEF relies on a pre-selected exposure as reference to detect inconsistent motion by computing structural consistency between the reference patch $\mathbf{s}_r$ and a co-located patch $\mathbf{s}_k$ from the k-th exposure

$$\rho_k = \mathbf{s}_r^T \mathbf{s}_k \approx \frac{\bar{\mathbf{x}}_r^T \bar{\mathbf{x}}_k + \epsilon}{\|\bar{\mathbf{x}}_r\| \, \|\bar{\mathbf{x}}_k\| + \epsilon}, \tag{3.14}$$

where ϵ is a small positive constant to ensure the robustness of the computation to sensor noise. We also use box filter [36] to calculate the structural similarity for the $\mathcal{O}(MK)$ implementation. Based on Eq. (3.14), K binary maps can be computed to identify static and dynamic regions with a pre-defined threshold T

$$\mathbf{B}_k(i) = \begin{cases} 1 & \text{if } \boldsymbol{\rho}_k(i) \geq T \\ 0 & \text{if } \boldsymbol{\rho}_k(i) < T, \end{cases} \tag{3.15}$$

where i denotes the spatial index. $\mathbf{B}_k$ is further refined with the help of the intensity mapping function (IMF) [75]. For our multi-scale fast SPD-MEF approach, it is straightforward to make the structural consistency measurements and generate the corresponding binary maps at each scale. For simplicity, we perform object motion detection at the original scale only. Finally, the dynamic regions are corrected by IMF [75] for multi-scale fusion.

Algorithm 1 Proposed multi-scale fast SPD-MEF method

Input: Registered source image sequence $\{\mathbf{X}_k\}$

Output: Fused image $\hat{\mathbf{X}}$

1: Select a reference image, detect motions via structural consistency and compensate the moving regions using IMF
2: **for** each Scale $j \in [1, J]$ **do**
3: Compute $\mathbf{L}_k^{(j)}$, $\boldsymbol{\gamma}_k^{(j)}$ and get the fused detail layer $\hat{\mathbf{D}}^{(j)}$
4: Downsample $\mathbf{L}_k^{(j)}$
5: **if** $j == J$ **then**
6: Compute $\boldsymbol{\alpha}_k^{(J)}$, $\boldsymbol{\gamma}_k^{(J)}$ and $\mathbf{L}_k^{(J)}$ and get the fused base layer and detail layer
7: **end if**
8: **end for**
9: Obtain the fused image $\hat{\mathbf{X}}$ by progressive upsampling and summing up

3.6 Experiments

In this section, we first present the implementation details of the proposed multi-scale fast SPD-MEF approach. Then we provide qualitative and quantitative results of our method against the state-of-the-arts along with ablation experiments for self-comparison. Last, we conduct theoretical and empirical computational complexity analysis. We summarize the proposed multi-scale fast SPD-MEF approach in Algorithm 1. Our method does not introduce any new parameter; the default parameters are inherited from previous publications [76, 62, 75], including the patch/mean filter dimension $N = 9 \times 9 \times 3$ from [76], $\lambda = 20$ that determines the arctan curve from [62], and $\epsilon = 0.03^2$ in Eq. (3.14) and $T = 0.8$ in Eq. (3.15) from [75].

3.6.1 Static scene comparison

We compare our method with nine MEF algorithms on 21 static scenes, including Mertens09 [82], Shen11 [106], Gu12 [27], Li13 [65], Shen14 [104], SPD-MEF [75], Nejati17 [89], GGIF [53], and Ancuti17 [1]. The fused images of all algorithms are either from the original authors or generated by the publicly available implementations with default settings.

Fig. 3.6 visually compares our method with existing MEF algorithms on the

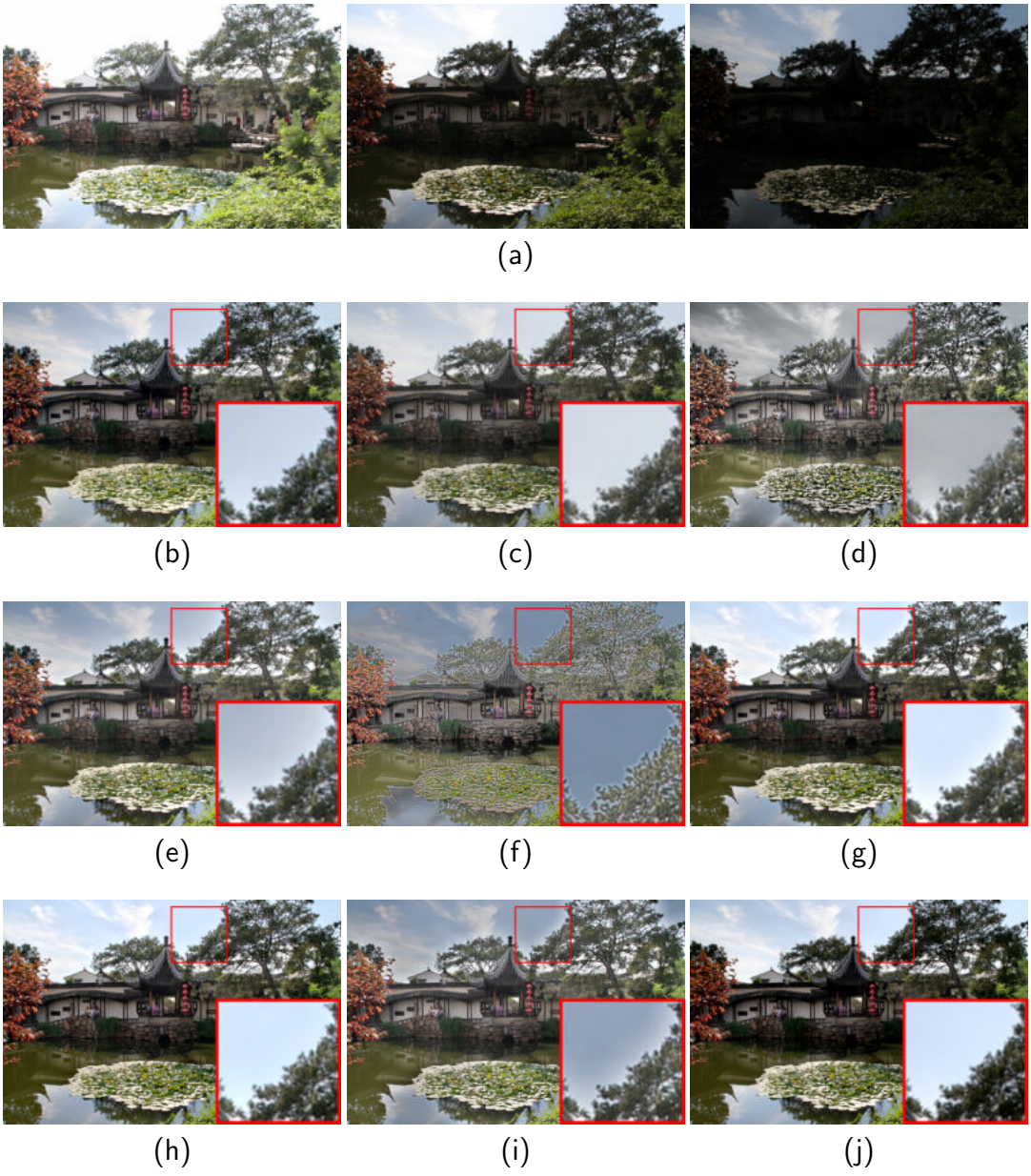


Figure 3.6: Visual comparison of our method with static MEF algorithms. (a) Image sequence "Chinese garden" (courtesy of Bartlomiej Okonek). (b) Mertens09 [82]. (c) Shen11 [106]. (d) Gu12 [27]. (e) Li13 [65]. (f) Shen14 [104]. (g) SPD-MEF [75]. (h) Nejati17 [89]. (i) Ancuti17 [1]. (j) Ours. The corresponding MEF-SSIM scores can be found in Table 3.1.

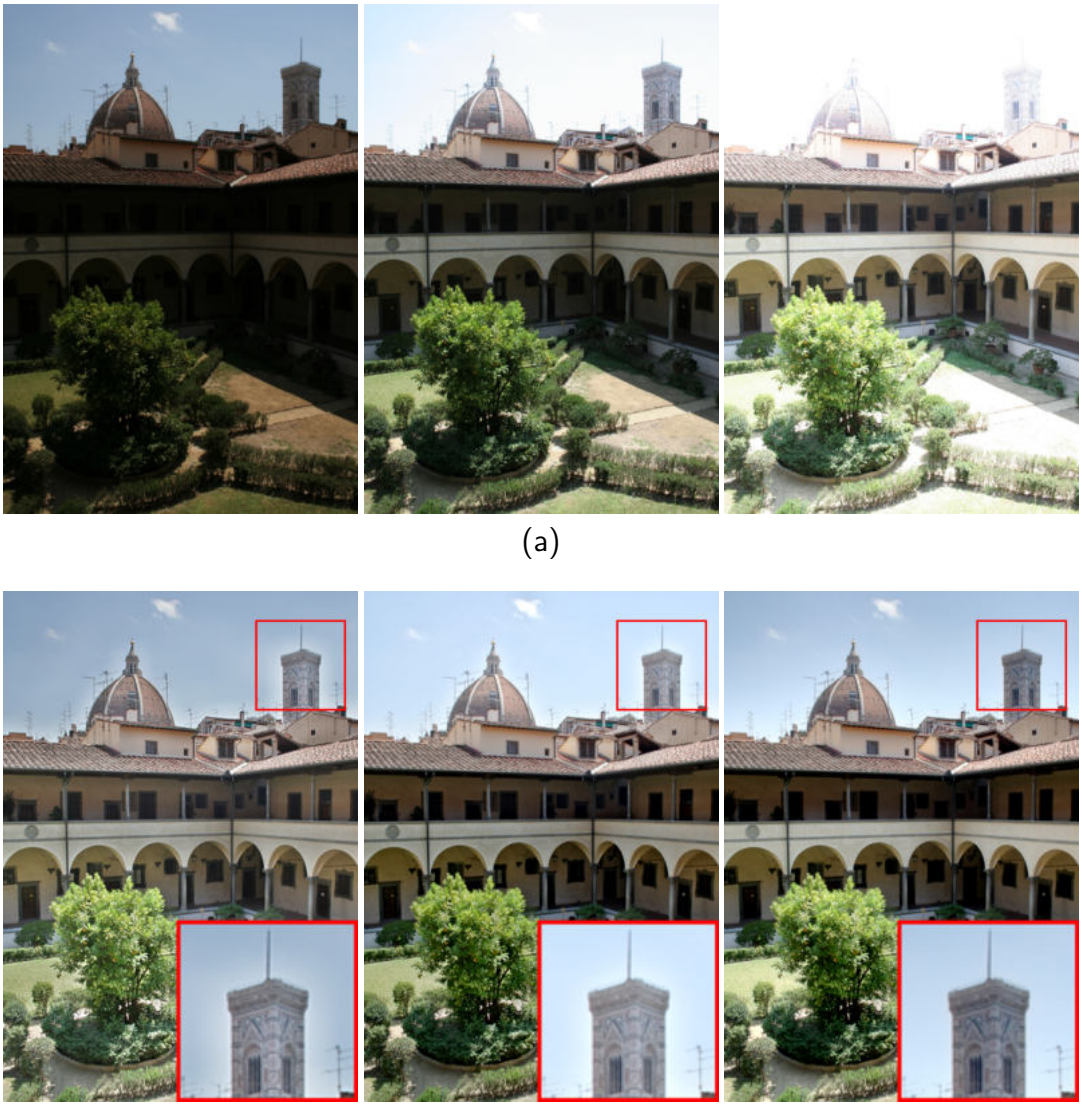


Figure 3.7: Example of halo artifacts. (a) Image sequence "Laurenziana" (courtesy of Bartlomiej Okonek). (b) Ancuti17 [1]. (c) SPD-MEF [75]. (d) Ours.

image sequence "Chinese garden". Although built upon Mertens09 [82], Shen14 [104] generates an unnatural appearance with annoying color and structure distortions due to nonlinearly enhancing the detail layer by a simple sigmoid function. Relying on the gradient information only, Gu12 [27] makes little use of color information, and over-shoots the details by solving the Poisson equation in gradient domain [22].

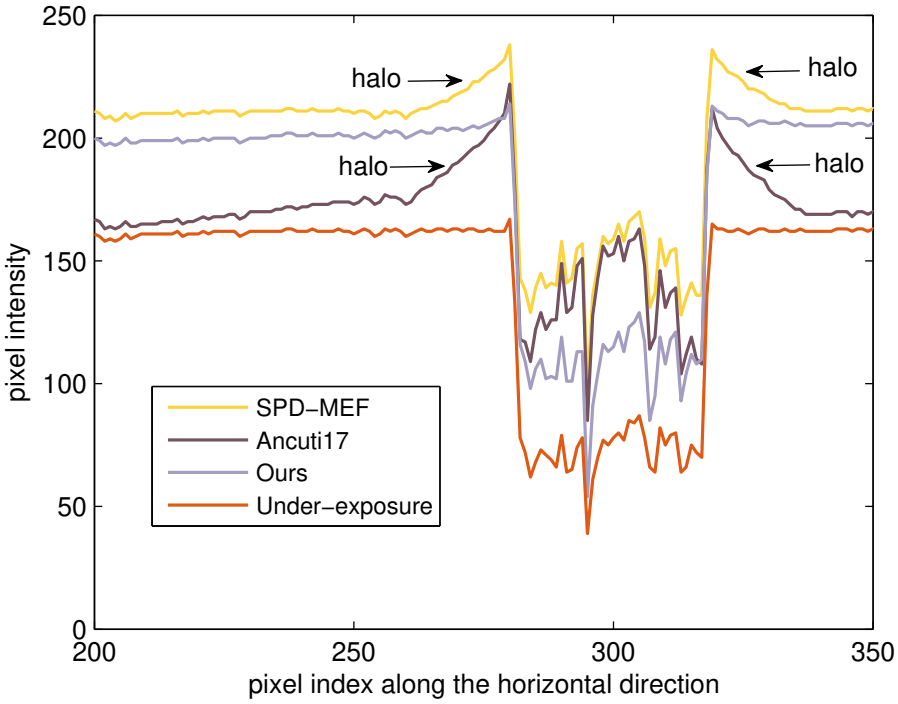


Figure 3.8: Pixel intensity analysis of the zoom-in patches in Fig. 3.7 along the horizontal direction. The patch from the under-exposure is used as reference since it has the best local quality. The halos generated by SPD-MEF [75] and Ancuti17 [1] are clearly seen as unwanted smoothing near the boundaries. Our method closely approximates the boundaries of the reference patch with an overall brighter appearance as expected.

The color appearance produced by Shen11 [106] is slightly better, but the overall contrast is somewhat reduced. In addition, ringing artifacts appear near strong edges because of excessive nonlinear manipulation of subbands. The above three methods equate detail enhancement with visual quality improvement, which is not always true, especially in the case of over-enhancement. Li13 [65], SPD-MEF [75], Nejati17 [89], and Ancuti17 [1] exhibit different degrees of halo artifacts in the sky regions, which are zoomed in for improved visibility. Compared to Li13, Nejati17 reduces the halos by replacing Gaussian filtering with guided filtering [36] in the two-layer decomposition. Severe halos are unavoidable for single-scale methods like SPD-MEF and Ancuti17 when they strike a balance between spatial consistency and

detail preservation. Mertens09 [82] and our method produce similar results on this sequence with little artifacts.

To better understand the emergence of halo artifacts in MEF, we show another visual example in Fig. 3.7, where we compare our method with SPD-MEF [75] and Ancuti17 [1] on the image sequence "Laurenziana". The boundaries (*e.g.*, zoom-in patches) between the foreground and the background with large dynamic range differences are the main sources of halo artifacts. To faithfully reproduce fine details across exposures, single-scale methods such as SPD-MEF and Ancuti17 cannot choose a kernel size that is too small (spatial inconsistency) or too large (detail loss) to compute local statistics for weighted fusion. This inevitably leads to unwanted smoothing of boundaries (see Fig. 3.8), which is visually perceived as "halos". The proposed multi-scale SPD-MEF approach resolves this issue by using a medium kernel to preserve details at each scale. The equivalent kernel size at the original scale is large enough to distribute such blurring more globally, which effectively suppresses the halos and makes the background brighter, as shown in Fig. 3.8.

We objectively evaluate the quality of fused images generated by different MEF algorithms using MEF-SSIM [78], which has been verified by comparing to human data [126] and through perceptual optimization [74]. MEF-SSIM [78] summarizes local structure preservation and global luminance consistency into an overall score between 0 and 1, with a higher value indicating better perceptual quality. The results are listed in Table 3.1, where we observe that our method achieves the best performance on average. Specifically, it outperforms the competing algorithms on 13 out of 21 natural scenes.

3.6.2 Dynamic scene comparison

On dynamic sequences, we compare our method with eight state-of-the-art HDR deghosting algorithms that cover a wide range of design philosophies, including low

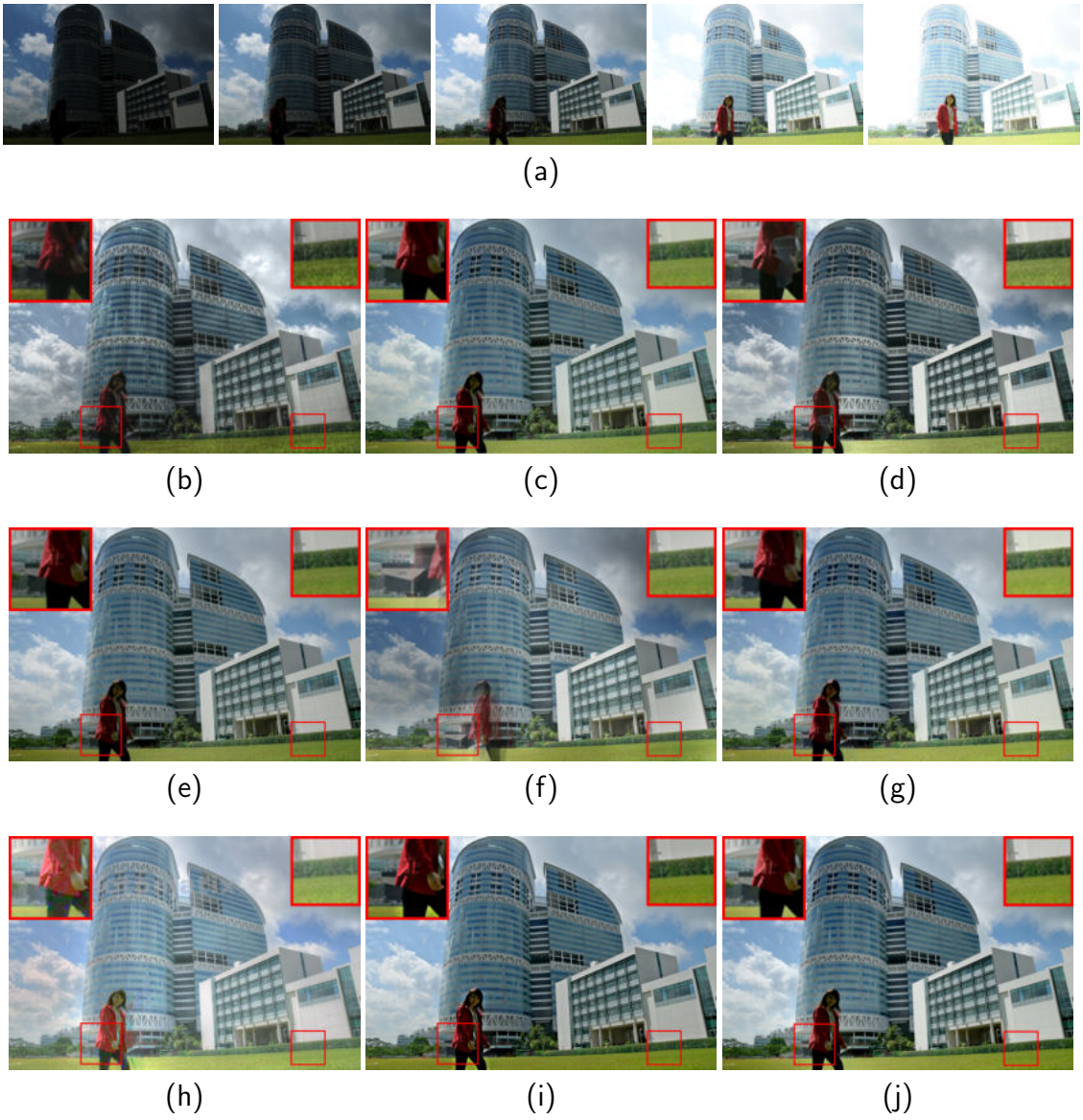


Figure 3.9: Visual comparison of our method with dynamic MEF algorithms. (a) Image sequence "Girl" (courtesy of Zhengguo Li). (b) Sen12 [102]. (c) Hu13 [39]. (d) Lee14 [60]. (e) Li14 [70]. (f) Liu15 [73]. (g) Qin15 [95]. (h) Oh15 [90]. (i) SPD-MEF [75]. (j) Ours.

Table 3.1: Quantitative comparison of our method with existing MEF algorithms using MEF-SSIM [78]. The score ranges from 0 to 1 with a higher value indicating better performance. The best results are highlighted in bold

Image sequence	Mertens09 [82]	Shen11 [106]	Gu12 [27]	Li13 [65]	Shen14 [104]	SPD-MEF [75]	Nejati17 [89]	GGIF [53]	Ancuti17 [1]	Ours
Arno	**0.991**	0.955	0.890	0.969	0.846	0.984	0.985	0.970	0.915	0.990
Balloons	0.969	0.940	0.913	0.948	0.776	0.969	**0.971**	0.951	0.929	0.963
Belgium house	0.971	0.935	0.896	0.964	0.709	0.973	0.972	0.968	0.938	**0.977**
Cave	0.975	0.946	0.934	0.978	0.788	**0.985**	0.979	0.979	0.958	0.984
Chinese garden	0.989	0.964	0.927	0.984	0.767	0.991	0.991	0.983	0.974	**0.994**
Church	0.989	0.959	0.866	0.992	0.878	**0.993**	0.991	0.992	0.980	0.991
Farmhouse	0.981	0.966	0.932	0.985	0.944	0.984	0.983	0.982	0.976	**0.986**
House	0.964	0.925	0.876	0.957	0.396	0.960	0.949	0.961	0.893	**0.973**
Lamp	**0.969**	0.917	0.875	0.929	0.539	0.956	0.960	0.945	0.877	0.967
Landscape	0.976	0.955	0.941	0.942	0.880	**0.993**	0.992	0.947	0.939	0.989
Laurenziana	0.988	0.956	0.873	0.987	0.881	0.987	0.986	0.985	0.957	**0.989**
Madison capitol	0.977	0.940	0.864	0.968	0.542	0.983	0.978	0.969	0.907	**0.990**
Mask	0.987	0.964	0.879	0.979	0.827	0.988	0.988	0.977	0.948	**0.991**
Office	0.985	0.958	0.900	0.967	0.756	**0.990**	0.988	0.984	0.957	0.989
Ostrow	0.974	0.950	0.877	0.967	0.786	0.978	0.978	0.977	0.925	**0.979**
Room	0.974	0.945	0.853	0.986	0.729	0.978	0.976	**0.983**	0.958	0.980
Set	0.986	0.974	0.911	0.960	0.873	0.988	0.988	0.966	0.905	**0.992**
Tower	0.986	0.946	0.932	0.986	0.779	0.986	0.986	0.986	0.962	**0.988**
Venice	0.966	0.930	0.889	0.954	0.765	0.984	0.976	0.952	0.932	**0.984**
Window	**0.982**	0.959	0.876	0.971	0.879	0.982	0.981	0.972	0.936	**0.982**
Yellow hall	0.995	0.983	0.869	0.990	0.866	0.995	0.996	0.987	0.966	**0.997**
Average	0.980	0.951	0.894	0.970	0.772	0.982	0.981	0.972	0.940	**0.985**

rank-based methods Lee14 [60] and Oh15 [90], energy-based methods Sen12 [102], Hu13 [39] and Qin15 [95], and feature-based methods Li14 [70], Liu15 [73] and SPD-MEF [75]. For HDR reconstruction algorithms (*i.e.*, fusion in radiance domain), the Debevec and Malik's method [10] is used to estimate the camera response function. In order to generate LDR images for visual comparison, Lee14 makes use of the MATLAB function tonemap(), and Sen12 and Hu13 fuse aligned LDR sequences using Photomatix [92] and Mertens09, respectively.

Fig. 3.9 shows the fusion results on the image sequence "Girl". Sen12 [102] produces an over-enhanced image that looks unnatural. This is largely attributed to the exaggerated settings of Photomatix [92] to enhance HDR details. In general, it is delicate for HDR reconstruction algorithms to select proper tone mapping operators to compress the dynamic range. Lee14 [60] and Oh15 [90] suffer from ghosting artifacts, which is expected because small and overlapping motion does not satisfy

Table 3.2: Computational complexity comparison of our method against state-of-the-art deghosting schemes

Algorithm	Complexity
Sen12 [102]	$\mathcal{O}(I_i NMK^2)$
Hu13 [39]	$\mathcal{O}(I_i N(M \log M)K)$
Lee14 [60]	$\mathcal{O}(I_o I_i MK^2)$
Li14 [70]	$\mathcal{O}(MK)$
Qin15 [95]	$\mathcal{O}(I_i NM^2K)$
Oh15 [90]	$\mathcal{O}(I_o I_i MK^2)$
SPD-MEF [75]	$\mathcal{O}(NMK)$
Ours	$\mathcal{O}(MK)$

Table 3.3: Average running time comparison on 12 dynamic scenes of approximately the same size ($683 \times 1024 \times 3 \times 3$)

Alg	Sen12 [102]	Hu13 [39]	Lee14 [60]	Qin15 [95]	Oh15 [90]	SPD-MEF [75]	Ours
Env	MATLAB+Mex	MATLAB+Mex	MATLAB+Mex	MATLAB+Mex	MATLAB	MATLAB	MATLAB
Time (s)	75.28 ± 20.48	114.96 ± 45.29	36.91 ± 11.55	465.06 ± 298.87	40.93 ± 9.93	57.48 ± 3.21	1.92 ± 0.20

the low rank assumption. In addition, solving such an optimization with only a limited number of exposures is relatively unstable, and may result in other forms of distortions. Liu15 [73] relies on dense SIFT features, which however may not be robust to exposure, making deghosting unsuccessful. Some halos around the girl's leg are visible in the fused image generated by SPD-MEF [75]. Hu13 [39] and Qin15 [95] may generate shifted colors and deformed structures due to inaccurate patch match during energy minimization. The results produced by Li14 [70] and our method are visually similar on this sequence.

3.6.3 Computational complexity comparison

We conduct a brief computational complexity analysis of HDR deghosting schemes in terms of the number of floating-point operations and refer the interested readers to [60, 75] for a more-detailed treatment. Assume the input sequence has K exposures, each of which contains M pixels ($K \ll M$); for patch-wise methods, the patch

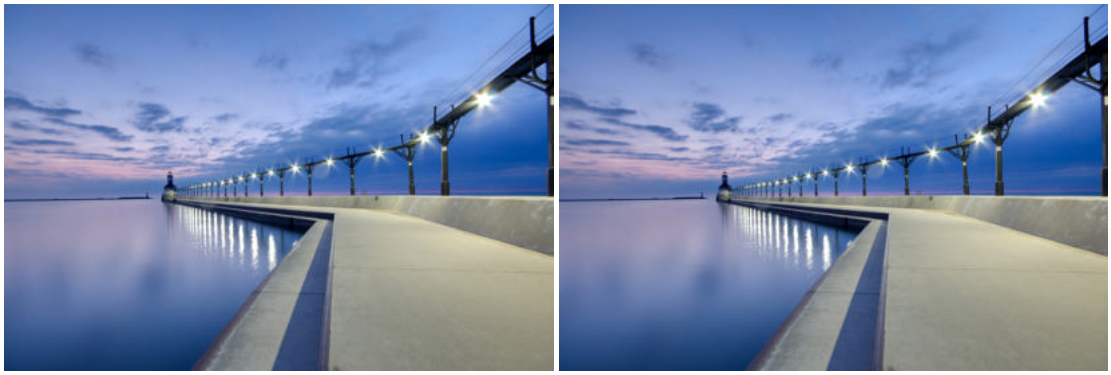
(a)

(b) (c) (d)

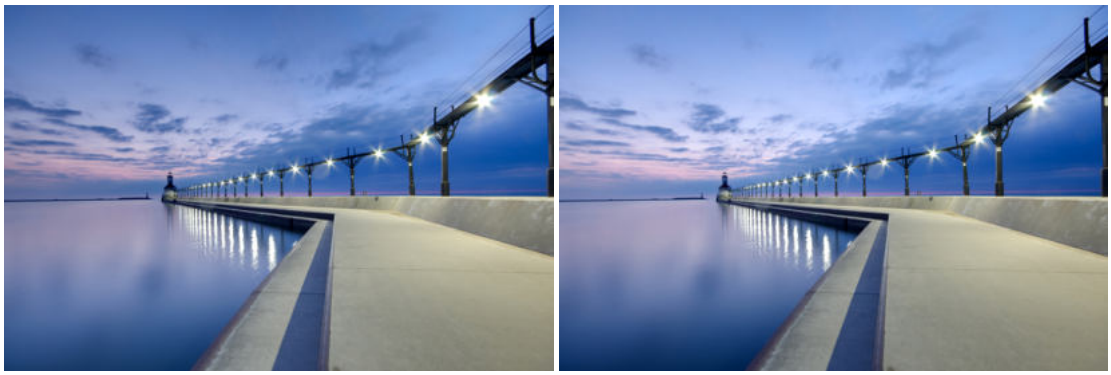
Figure 3.10: The number of scales in our method plays an important role in the visual quality of fused images. (a) Image sequence "Balloons" (courtesy of Erik Reinhard). (b) Single-scale result with an MEF-SSIM of 0.851. (c) Three-scale result with an MEF-SSIM of 0.926. (d) Five-scale result with an MEF-SSIM of 0.963, whose scale is computed adaptively using Eq. (3.12).

dimension is assumed to be N; for iterative algorithms, the iteration numbers used in the inner and outer loops are I_i and I_o, respectively. The analysis results are listed in Table 3.2, where we find that the proposed method and Li14 [70] enjoy the lowest computational complexity, which is linear with the number of pixels in the sequence. The average running time of different algorithms on 12 natural scenes of approximately the same size is also listed in Table 3.3. The experiment is conducted on a computer with 4G Hz CPU and 32G RAM. The make a fair comparison, the stride of SPD-MEF is set to one instead of the default two. Our MATLAB code runs the fastest among the competing algorithms that demonstrate satisfactory performance in deghosting, and accelerates the original SPD-MEF more than 30 times. When compared to Mertens09 [82] that is widely adopted in mobile devices as a core module to capture HDR-like pictures (*i.e.*, the HDR mode) [35], our method shares the same computational complexity, and therefore has great potentials in enabling

(a)



(b) (c)



(d) (e)

Figure 3.11: Visual comparison of different intensity weight functions. (a) Image sequence "Set" (courtesy of Jianbing Shen). (b) Fused image by the hat-shaped curve with an MEF-SSIM of 0.985. (c) Fused image by the Gaussian curve with an MEF-SSIM of 0.983. (d) Fused image by the Bell-shaped curve with an MEF-SSIM of 0.980. (e) Fused image by the proposed intensity weight function with an MEF-SSIM 0.992.

real-time mobile applications for challenging dynamic scenes.

3.6.4 Ablation experiments

Impact of the number of scales

We analyze how the number of scales J affects the fusion performance by using the image sequence "Balloons". When the number of scales increases, our method gradually spreads the halos around the two balloons over the background, making the sky brighter and perceptually more appealing (see Fig. 3.10). The spatial inconsistency is also effectively reduced at the price of some detail loss (*e.g.*, around the sun). Our adaptive strategy of selecting the highest scale J according to Eq. (3.12) keeps a good balance among spatial consistency, detail preservation, and halo suppression.

Impact of the intensity weight function

The desired base layer corresponds to the primary dynamic range, and it is blended based on well-exposedness measures. Here we visually compare four such measures (see Fig. 3.4), among which the Gaussian curve and its variants [82, 75] have been widely used to construct weights in MEF. From Fig. 3.11, we find that the hat-shaped and Gaussian curves generate visually close results and MEF-SSIM values because both of them weight intensities in a similar fashion. Compared to the bell-shaped curve, the proposed weight function is more friendly to less well-exposed intensities, resulting in a slightly brighter overall appearance with a higher MEF-SSIM value.

3.7 Conclusion

In this chapter, we studied structural patch decomposition (SPD) for MEF and showed that an unnormalized approximation of SPD-MEF is closely related to previous MEF schemes. The relationship with pixel level fusion and two layer decomposition are analysed in detail. This insight allows us to avoid performing SPD

explicitly, which speeds up SPD-MEF more than 30 times. We then made SPD-MEF multi-scale, which effectively reduces halo artifacts near strong edges. The impact of intensity weight function and decomposition level are illustrated. Extended experiments indicate the effectiveness of the method in both static and dynamic scene. The proposed multi-scale fast SPD-MEF approach provides a practical solution for mobile applications with high resolution input images.

Chapter 4

Deep Multi-exposure Image Fusion

MEF is a widely used approach to high dynamic range imaging. In this chapter, we investigate the effectiveness of convolutional neural network for MEF. First, we exploit MEF using CCN features extracted via a trained network given that the selection of features for fusion weight calculation is important to the performance of MEF. Both the selection of network and the selection of convolution layer are studied. With the extracted CNN feature map, we compute the local visibility and consistency maps to determine the weight map for MEF. The proposed method works well for both static and dynamic scenes. It exhibits competitive quantitative measures, and presents perceptually pleasing MEF outputs with little halo effects. Second, we use explore the end-to-end training for MEF. The network can produce pleasing fused results in static scene. Due to the lack of dynamic scene data, we the network can introduce some ghosting effect. How to establish a dynamic MEF dataset with ground-truth is a meaningful topic worth further investigation.

4.1 Introduction

Natural scenes usually span a high dynamic range, which is challenging for digital single-lens reflex cameras to capture. High dynamic range imaging aims to extend the dynamic range of digital cameras by capturing an image sequence with multiple

exposures. There are two categories of methods to generate an HDR-like image: multi-exposure fusion [82, 76, 75] in image domain, and HDR content reconstruction via camera response function estimation and tone mapping [77, 62]. Due to the complexity of recovering CRF and designing tone mapper [62], MEF is more preferably used in most consumer grade devices such as camera phones and digital cameras for its simpler implementation.

Many works on MEF have been reported in past decades. In [25], the source images are segmented into non-overlapped patches and fused based on the rule of max-entropy. The blocking artifact is avoided by blending neighboring patches. In [76], an image patch is decomposed into three components: strength, structure and intensity, and the fusion is performed on each component. This scheme is extended [76, 75] to fuse dynamic scenes via calculating the structural similarity between corresponding spatial patches. The MEF method [82] computes the weight map via three image quality measures (contrast, color saturation and well-exposedness) and fuses the images in an efficient multi-resolution framework. However, this method only works for static scenes. Zhang *et al.* [133] proposed a gradient based MEF method for both static and dynamic scenes. The gradient magnitude is used for setting the fusion weight and the gradient direction is used to detect the moving objects across the sequence. Similarly, Gu *et al.* [27] utilized the structure tensor in gradient domain to compute the weight. In [73], Liu *et al.* proposed a SIFT descriptor based MEF method, which exhibits superior performance to [133] due to its more effective feature representation. In single-scale fashion, the edge-preserving filters [96, 65, 64, 52] are used for smoothing weight maps to alleviate artifacts.

CNN has achieved state-of-the art results in high-level vision problems [124], such as classification, segmentation and object detection, owing to its powerful discriminative feature learning ability. Recently, CNN has also been successfully used in many low-level vision problems such as super-resolution, denoising, and enhance-

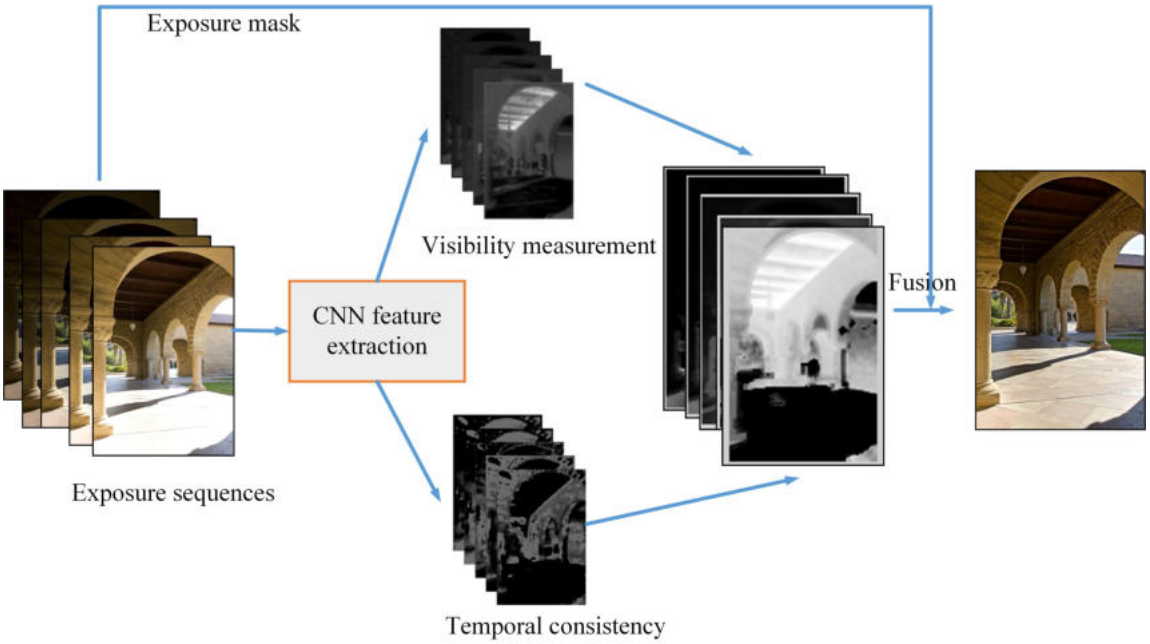


Figure 4.1: Flowchart of the proposed CNN feature based multi-exposure fusion method.

ment [127, 48, 41, 7], leading to impressive outcomes. Guided by the labeled training samples, deep CNNs can extract more effective features than conventional hand-crafted features.

However, few works have been reported to apply CNN for MEF. This is mainly because of the lack of ground-truth output for a given multiple exposure sequence so that the end-to-end learning of CNN cannot be adopted. In our work, we exploit the features of pre-trained CNNs to compute weight maps for simple yet effective MEF, which works for both static and dynamic scenes. Besides, we explore to train a CNN to generate the MEF output by use of two similar datasets.

4.2 Multi-exposure fusion with CNN features

Considering the fact that there are no strict ground-truth images in MEF to train an end-to-end CNN, we adopt the pre-trained networks in other tasks to extract the

feature. In this section, we exploit the possibility of utilizing convolutional neural network (CNN) [110, 127, 48] features for MEF, which is the first of its kind to the best of our knowledge. We found that shallow-layer CNN features can be leveraged for both static and dynamic MEF, achieving state-of-the-art MEF performance. This section presents the proposed CNN feature based MEF method. The flowchart of our method is shown in Fig. 4.1, which contains four major components: CNN feature extraction, pixel visibility measurement, temporal consistency check, and exposure mask calculation.

4.2.1 Related works

Feature extraction plays a pivotal role in determining the weight map for MEF. Most MEF methods [82, 27, 133] implicitly or explicitly incorporate the feature extraction in the design of fusion weights. For example, the second order Laplacian filter, one-order gradient operator and SIFT descriptor are used as the feature extractors in [82, 133, 73], respectively. All the features used in current MEF methods are hand-crafted features.

4.2.2 Feature extraction and visibility measurement

In general, there are two types of deep CNNs available for feature extraction: regression network (usually for low-level vision problems) and classification network (usually for low-level vision problems). In our application, dense features are required because we need to compute a weight at each pixel. For regression networks such as the denoising network [127] and super-resolution network [48], usually dense features can be obtained at each layer. For classification networks such as the VGG-Net [110], the dense features can only be obtained at shallow layers, which are however guided by the deeper layer sparse features with certain high level semantic information (e.g., discriminative parts, class labels). The selection of network and layer will be dis-

cussed on Section 4.2.5.

Let $I_i, i = 1, 2, 3, ..., K$, be a sequence of K multi-exposure images. The feature map of each source image is extracted as

$$\mathbf{F}_i(x, y) = CNN(I_i)(x, y), \tag{4.1}$$

where $CNN(\cdot)$ is a pre-trained deep network at some layer. For one pixel (x, y), we can obtain a feature vector whose dimension is the number of filters used in that convolutional layer. The response of a pixel to CNN filters can indicate whether this pixel is important or informative for image representation. Therefore, the strength of feature vector $V_i(x, y)$ can be used to determine the visibility of pixel $I_i(x, y)$. We measure the visibility of pixel $I_i(x, y)$ as the L_1 norm of $V_i(x, y)$:

$$V_i(x, y) = \|\mathbf{F}_i(x, y)\|_1. \tag{4.2}$$

Compared with L_2 norm, the L_1 norm is selected due to its simpler calculation. The degree of visibility will affect value of weight to be assigned to this pixel. The pixels that have good local contrast are usually given bigger weights.

4.2.3 Temporal consistency

When the source images are taken in dynamic scenes, motion detection is needed to avoid the ghosting effect. In our framework, the motion detection can be easily implemented via calculating the Euclidean distance of two normalized feature vectors. Denote by $\bar{\mathbf{F}} = \frac{\mathbf{F}}{\|\mathbf{F}\|}$ the L_2 normalized feature of $\mathbf{F}(x, y)$ to remove the impact of exposure difference. The distance of two feature vectors at a pixel of two images is computed as:

$$s_{ij}(x, y)^2 = \|\bar{\mathbf{F}}_i(x, y) - \bar{\mathbf{F}}_j(x, y)\|_2^2, \tag{4.3}$$

$s(x, y)^2$ measures the similarity of two vectors. A smaller $s(x, y)^2$ represents stronger temporal consistency. When there is motion, the consistency will be destroyed,

resulting in a smaller similarity value. We use a Gaussian kernel to map the similarity between $\bar{\mathbf{F}}_\mathbf{i}(\mathbf{x}, \mathbf{y})$ and $\bar{\mathbf{F}}_\mathbf{j}(\mathbf{x}, \mathbf{y})$ into the range of [0, 1]:

$$S_i(x, y) = \sum_{j=1}^{K} exp\frac{-s_{ij}(x, y)^2}{2\sigma^2}, \tag{4.4}$$

where σ is a constant and is set as 0.05 here. A larger weight should be given to the pixels which are temporally consistent.

4.2.4 Fusion

With the visibility and similarity weight maps V and S, we can get the final weight map $W_i(x, y)$ as follows:

$$W_i(x, y) = \frac{V_i(x, y) \times S_i(x, y) \times M_i(x, y)}{\sum_{j=1}^{K} V_i(x, y) \times S_i(x, y) \times M_i(x, y) + \alpha}, \tag{4.5}$$

where $M_i(x, y)$ is the exposure mask which is computed on the pixel intensity. To avoid division by zero, we add a small coefficient α with value 10^{-10}. The widely used mask is the hat function defined as follow:

$$M_i(x, y) = \begin{cases} 1, & \beta < I_i(x, y) < 1 - \beta, \\ 0, & else, \end{cases} \tag{4.6}$$

where $\beta \in [0, 1]$ is a parameter controlling the exposure quality when the input images are normalized. It can effectively remove the poor exposure pixels. We choose β as 0.2 in our implementation. In practice, the weight map W can be smoothed by the edge-aware recursive filter [64] to further reduce the halo effect. Finally, we fuse the images as follows to produce the MEF output I_f:

$$I_f(x, y) = \sum_{i=1}^{K} I_i(x, y) \times W_i(x, y). \tag{4.7}$$

Table 4.1: The MEF-SSIM scores by different networks at different layers on static scene dataset [76]

MEF-SSIM: Network Type \ Feature Layer	1	3	10	18
Denoising [127]	0.869	0.970	0.969	0.965
Super-resolution [48]	0.867	0.957	0.846	0.930
Classification [110]	0.969	0.620	0.610	0.560

4.2.5 Experimental results

In this section, we discuss the selection of CNN networks and the associated layers for feature extraction. Then we compare the performance of CNN features with traditional hand-crafted features. Finally, we compare our method with state-of-the-art MEF methods. In all our experiments, we adopt the metric MEF-SSIM proposed in [79] as a quantitative measure to evaluate the performance of MEF methods on static scenes.

The selection of network and layer

We adopt two regression networks, denoising network DnCNN [127] and super-resolution network VDSR [48], and one classification network VGG19 [110], in the experiments. By using the features at different layers to compute the weight map W, the MEF image can be computed and the MEF-SSIM scores of different networks on the static scene dataset [76] are listed in Table 4.1. Note that for the VGG19 network, the features at deeper layers become sparser due to pooling, and we interpolate the feature maps to calculate the weight for each pixel.

We can have the following observations from Table 4.1. First, the regression networks DnCNN and VDSR achieve their best MEF-SSIM indices at shallow layers (more specifically layer 3) but not the first layer. The deeper layers become more task specific for denoising and super-resolution, but cannot bring benefit for MEF.

Table 4.2: The MEF-SSIM scores of CNN and traditional features on the static scene dataset [76]

Feature type	CNN	SIFT	Gabor
MEF-SSIM	0.969	0.952	0.900

Second, the classification network VGG19 achieves its best MEF-SSIM index at the first layer. With the increase of layers, the performance decreases rapidly. This is because the feature maps are getting sparser and sparser due to the spatial pooling in VGG19 network so that the weight map becomes less accurate. On the other hand, though the layer 1 features are very shallow features, they are guided by the deeper high level semantic features in training, which can still capture information of image structures. This is one advantage of classification networks over regression networks. Third, layer 3 DnCNN features, layer 3 VDSR features and layer 1 VGG19 features achieve very similar MEF-SSIM indices (the visual quality of their fused images is also similar). Considering that layer 1 features need much less computational cost and storage space, we select layer 1 VGG19 features as our feature extractor.

Comparison between CNN and traditional features

We then compare the effectiveness of layer 1 VGG19 features with traditional Gabor features and SIFT features. The objective MEF-SSIM indices by the three types of features are shown in Table 4.2. It can be seen that CNN features bring much better MEF performance than the hand-crafted features. Due to the limit of space, we do not show the visual comparison here, while CNN features indeed bring better perceptual quality of MEF images.

Comparison with state-of-the-art methods

We then compare our method with state-of-the-art MEF algorithms. On static sequences, we compare it with "Ma" [75], "Mertens" [82], "Gu" [27], "Shutao" [65],

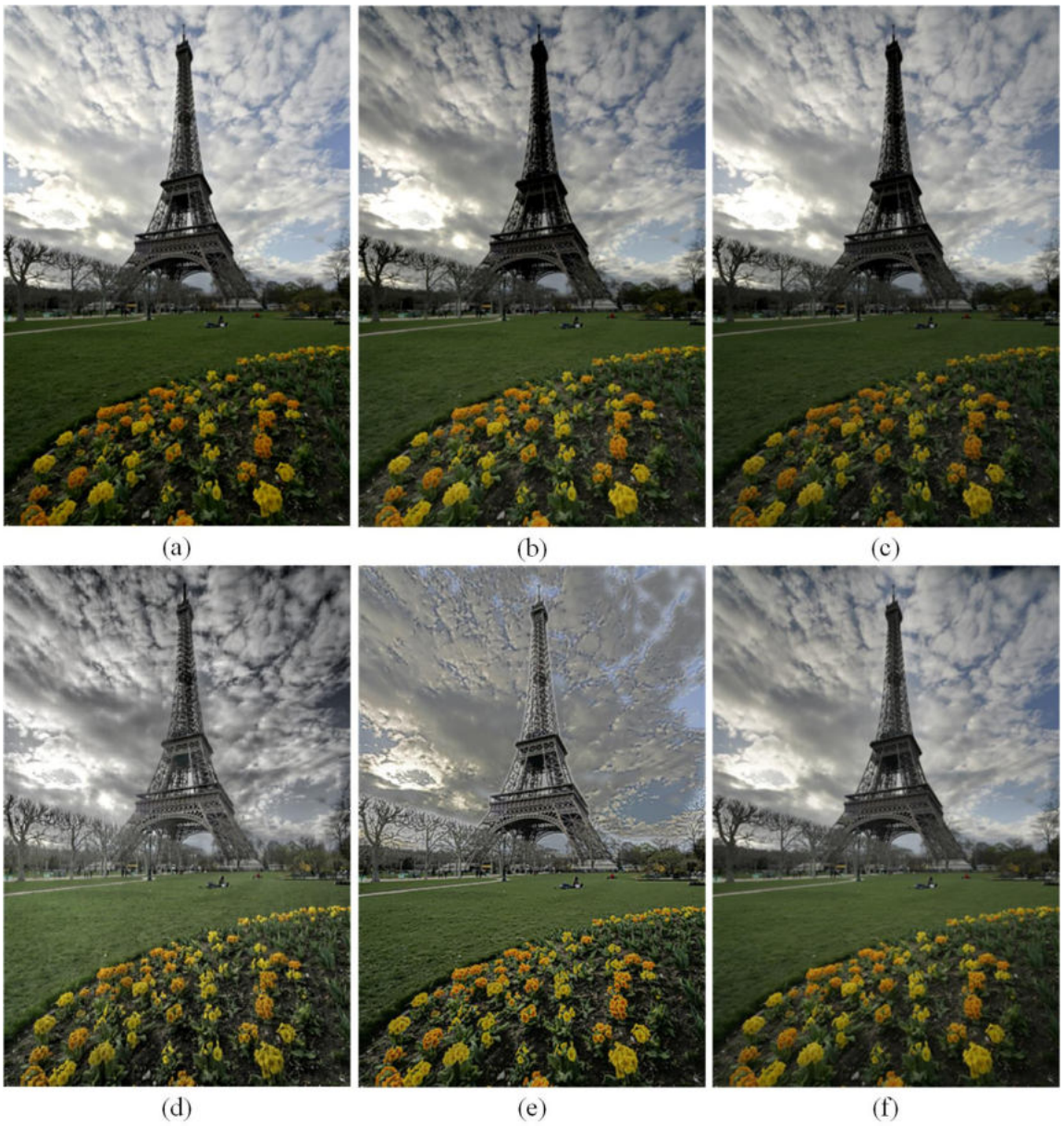


Figure 4.2: The MEF results by competing methods on a static scene. From (a) to (f): results by "Ma" [75], "Mertens" [82], "Gu" [27], "Shutao" [65], "Shen" [104], and "Ours".

Figure 4.3: The MEF results by competing methods on a dynamic scene. From (a) to (f): results by "Gallo" [24], "Li" [71], "Ma" [75], "Photomatix", "Sen" [102], and "Ours".

Table 4.3: The average MEF-SSIM scores of different methods on the static scene dataset [76]

Methods	[27]	[69]	[65]	[96]	[64]	[104]	[82]	[75]	Ours
MEF-SSIM	0.910	0.944	0.965	0.852	0.960	0.753	0.975	0.977	0.969

"Shen" [104], "Raman" [96] and "Li" [69]. Since some methods cannot be applied to dynamic sequences with motion, we compare our method with "Gallo" [24], "Li" [71], "Ma" [75], "Photomatix", and "Sen" [102] on dynamic data. The codes of competing methods "Ma" [75], "Mertens" [82] and "Sen" [102] are from the original authors and we use their default settings. Other MEF results are copied from [76] or from the original papers. The "Photomatix" is commercial software from the website [1]. The MEF-SSIM indices by competing methods on the static scene dataset [76] are listed in Table 4.3. Fig. 4.2 and Fig. 4.3 compare the MEF results by representative methods on a static scene and a dynamic scene, respectively. Please note that by far there is not a reliable objective quality measure for MEF results on dynamic scenes yet.

From Table 4.3, one can see that our method produces very competitive MEF-SSIM measures. Its average MEF-SSIM index is only lower than methods [82] and [75]. It should be noted that how to design a faithful objective quality measure for MEF is still a challenging issue. We found that some images with high MEF-SSIM scores exhibit obvious unpleasing artifacts. The average run-time of our method is 0.90s, comparable with [82] (0.87s), but much faster than [75] (2.33s) on a computer with 4G Hz CPU and 32G RAM. The run-time of other methods is not available due to the lack of source codes.

Fig. 4.2 compares the MEF results on a static scene. The methods "Ma" [75], "Mertens" [82] suffer from the detail loss in bright regions, as well as the low contrast in dark regions. Method "Shen" [104] and "Gu" [27] shows obvious over-enhancement. Though method "Shutao" [65] shows overall good visual quality on

[1] `https://www.hdrsoft.com//`

this image, it exhibits slightly lower contrast compared with our method. For the dynamic scene in Fig. 4.3, method "Gallo" [24] suffers from obvious color artifacts; methods "Li" [71] and "Ma" [75] lose some details in the roof of the arch. Method "Photomatix" exhibits severe over-exposure problem. The result by the tone mapping method "Sen" [102] is not natural with over-artistic effect. Our method provides a good balance between deghosting and detail preserving, proving the effectiveness of CNN feature on motion detection in multi-exposure sequence.

4.3 End-to-end learning for multi-exposure fusion

Although the CNN features extracted via a pre-trained network as described above can produce decent fusion results, these pre-trained networks are not specially trained for MEF tasks. In this section, we explore an end-to-end MEF algorithm by pre-processing two datasets. The original datasets are specially collected for low-light image enhancement and HDRI in radiance domain. In this section, we try to use them for static and dynamic MEF, respectively.

4.3.1 Related works

Recently, Prabhakar *et al.* [94] proposed two ways for CNN based MEF. One is to select the results produced by two representative MEF methods as the "ground-truth". The other is to learn the CNN by optimizing a no-reference image quality metric defined in [79]. Although the authors claimed an un-supervised learning via the quality metric [74], they actually used the optimized result in [74] as "ground-truth" for supervision. Besides, this method is not applicable to dynamic scenes. Cai *et al.* [7] built a extensive multi-exposure image dataset with ground truth by a subject study for low-light enhancement. Despite the improved detail enhancement in dark area, the result suffers serious noise. Regarding HDRI in radiance domain, Kalantari *et al.* [45] established a raw format multi-exposure image dataset for HDRI.

The ground-truth is defined by a static scene fusion in radiance domain. With the static scene as a reference frame, the under-exposure and over-exposure inputs are captured with human mobility. The limited scenes places restrictions on general scene HDRI including complex object motions.

4.3.2 Dataset

With the lack of MEF dataset with ground-truth, few works are reported for direct end-to-end MEF. The difficulty lies in the definition of ground-truth for MEF. We try to use the Cai's dataset for an end-to-end training for static MEF. The dataset contains large number multi-exposure sequences; however, the images in many sequences are not registered. Camera shake and moving objects exist in many scenes. The problems do not have a big influence on single image enhancement [7], which is one-one task. But for many-one task like MEF, non-registration and motion can bring strong distortion and blur. The dataset cannot be directly used to address our issue. To make it applicable, we delete the sequences with visible motion and non-registration. We screen 80 image sets for training and 20 for testing.

In terms of MEF in dynamic scene, there exists no direct available MEF dataset with ground truth. We use the similar dataset [45] as our training data through preprocessing. The collected input data is camera raw data with high bit, and label data is HDR data. We compress the inputs via Photoshop to 8-bit images. The output data is tone mapped by several typical tone mapping operators from Photomatix. We select the result with best performance as output via a coarse screening. The dataset contains both object motions and camera motions. It is difficult for a single network to learn to handle these two types of motions. We align the input using the method in [75] to reduce the burden of learning. After deleting some undesirable image sets suffering from severe noise in the processing of transforming to 8-bit images, we obtain 80 image sets. The collated dataset is divided into 70 training sets

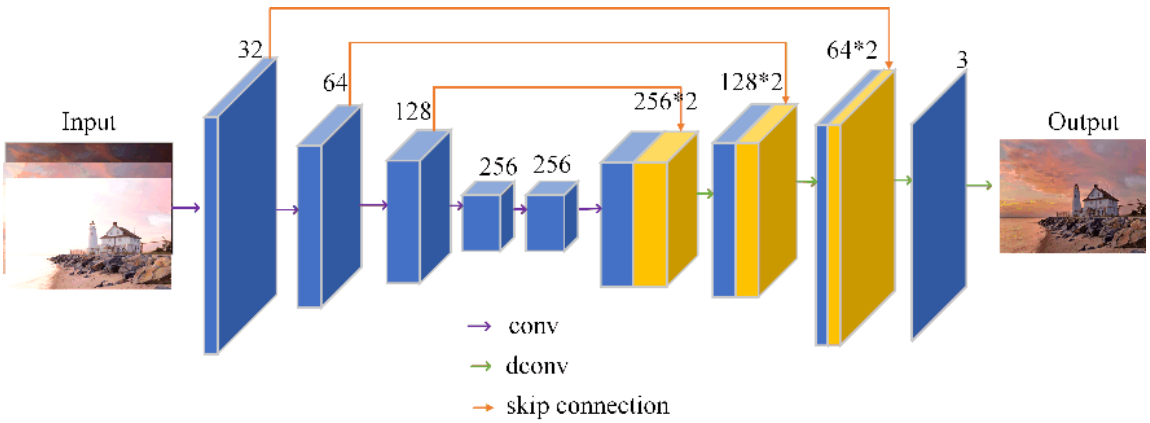


Figure 4.4: Flowchart of the proposed end-to-end multi-exposure fusion method.

and 10 testing sets.

4.3.3 Network architecture and training

We use U-Net architecture as our network as shown in Fig. 4.4, which is a typical encoder and decoder architecture. This framework has obtained great success in many high-level issues [100]. It can reduce the parameters and ensure large receptive field without a very deep network layer. With skip connections, the local and global information are fully intergraded, which can overcome the barriers of limited samples in MEF issue. We do not add the layer of explicit feature maps weighting [94, 63], while we implicitly do the merging by setting the number of filters. Different exposures share the same encoding stage for reduced parameters and speed-up. The input image number is set as three for simplicity in our implementation, since most HDR applications use three images as input. Larger number of input images can be easily extendable.

The most used loss function is MSE loss. However, we found MSE loss can lead to detail and contrast loss through our experiment. We use L_1 loss as our loss function, which can generate more edge sharped results. Given a training set

Table 4.4: The average MEF-SSIM scores by different methods on the static dataset [76]

Methods	[27]	[69]	[65]	[96]	[64]	[104]	[82]	[75]	Ours
MEF-SSIM	0.910	0.944	0.965	0.852	0.960	0.753	0.975	0.977	0.973

$\{I^i_{Input}, I^i_{Reference}\}_{i=1}^N$, the pixel-wise L_1 loss can be calculated as:

$$L^{l_1}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|F(I^i_{Input}) - I^i_{Reference}\|_1 \tag{4.8}$$

Where Θ stands for network parameters. Compared with MSE loss, L_1 loss can produce more edge-sharped results at the controllable cost of over-enhancement and artefacts.

Instead of training the whole images, we crop the image by patch size of 256×256 with stride of 64. The flipping and rotation operation are conducted for data augmentation. We use about 180,000 patches for training static MEF and 140,000 for dynamic MEF tasks, respectively. The learning rate is initialized as 0.1 and then decreases by a factor of 10. The batch size is 8 and we run 1000 epochs. The kernel size is 5×5. The stride is 2 and 1/2 in the encoding and decoding stage, respectively. The channel numbers are shown in Fig. 4.4. Each layer contains a 5×5 Convolution (Conv) and Rectified Linear Units (ReLU), except for the last layer. We train different networks for static and dynamic scene using the dataset mentioned above. To reduce the effect of color shift, we only train the luminance information. The color information is weighted via the method in [94].

4.3.4 Experimental results

A number of experiments have been made to indicate the effectiveness the trained end-to-end network. To objectively evaluate the quality of fused images and the performance of fusion algorithms, we employ MEF-SSIM [78] inherited from SSIM [117].

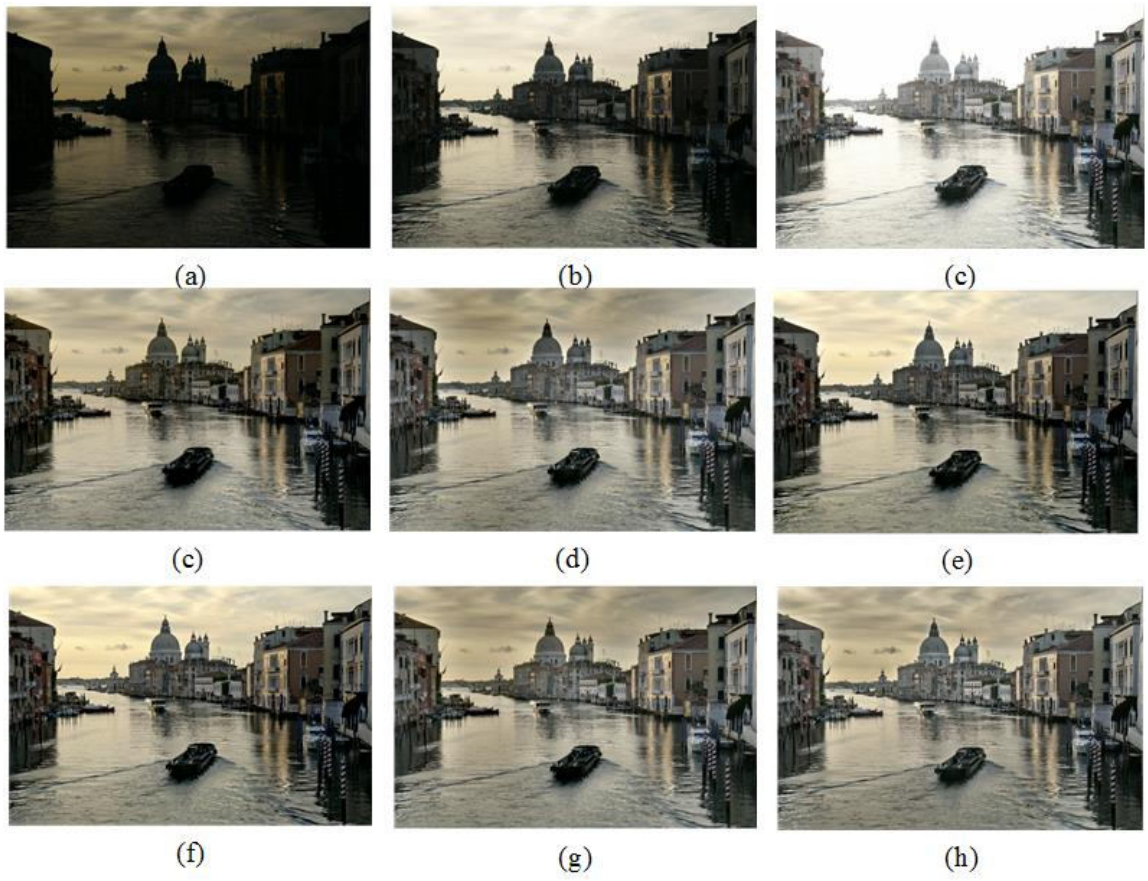


Figure 4.5: Visual comparison of our method with static MEF algorithms on a general static scene. (a)-(c): Exposure sequence. (d) Mertens09 [82]. (e) Li13 [65]. (f) SPD-MEF [75]. (h) Nejati17 [89]. (i) GGIF [53]. (g) Ours.

It can measure local structure preservation and global luminance consistency. A bigger value of MEF-SSIM score ranging from 0 to 1 indicates better quality. The MEF-SSIM scores by competing methods on the static scene dataset [78] are listed in Table 4.4. From Table 4.4, it can be observed that our method can gain quite competitive MEF-SSIM scores. The average MEF-SSIM score is comparable with the representative method [82].

A static example is given in Fig. 4.5, where we compare our method with 5 state-of-the-art MEF algorithms on 21 static scenes, including Mertens09 [82], Li13 [65], SPD-MEF [75], Nejati17 [89], and GGIF [53]. The fused images of all algorithms are

Figure 4.6: The testing result on a dynamic test scene. (a)-(c) Exposure sequence. (d) The ground-truth (c) Our test result.

either from the original authors or generated by the publicly available implementations with default settings. From Fig. 4.5, we can see that the result by the proposed end-to-end network are visually competitive with state-of-the-art traditional methods. Deep network can be applied to MEF, but the result is not evidently better than traditional methods.

A dynamic result in test set is given in Fig. 4.5. The result appears obvious ghosting artefact as shown in the labelled red box. The limited number of dynamic images and small proportion of moving background can not well detect the object motion, leading to ghosting appearance. Besides, the dynamic dataset only includes two exposure internals, which impedes the generalization ability on general dynamic scene fusion. It should be noted that some works [119, 45] based on the same dynamic dataset reported better deghosting performance. It is mainly because that they use decoded raw data are nearly linear with exposure time, which conducive to

mooting detecting.

4.4 Conclusion

This chapter made the first attempt to exploit the CNN features for weight design in MEF. Pre-trained CNN networks on other tasks were used to extract the features of each image. With these CNN features, the visibility and temporal consistency of each pixel in each image were defined, based on which the weights can be computed for MEF. We investigated the performance of regression networks and classification networks, and found that the very shallow layers of classification network can lead to desirable MEF outputs with low cost. Overall, the proposed method is simple and efficient to implement, and presents competitive results with state-of-the-arts on both static and dynamic scenes. Besides, we explore an end-to-end method for EMF. The obtained results are satisfactory in static scene, but suffer from ghosting effect in dynamic scene. How to build a trainable dataset for dynamic scene is a challenging issue.

Chapter 5

Real-world Image Super-resolution

Convolutional neural networks have been dominantly used in the field of single image super-resolution. However, most of the existing CNN models assume that the low-resolution images are produced by a simple degradation, more specifically, bicubic downsampling, from their HR counterpart. Unfortunately, the practical degradation of LR images can be far more complicated. The CNN models trained by simulated data become much less effective in real-world image super-resolution, despite the enormous efforts made in the design of network architectures and loss functions. To improve the performance of CNN in real-world SISR, we develop a novel dataset where the paired images on the same scene are captured by adjusting the lens focus of the digital camera. Image registration operations are conducted to crop the registered training pairs from the collected data. A plain regression network with simple loss functions are used to train a CNN model, which however generates exceptional SISR results on images either from our dataset or outside our dataset. Compared with those models trained by simulated data, our model can more effectively enhance the sharp edges and fine textures, and has better generalization capability.

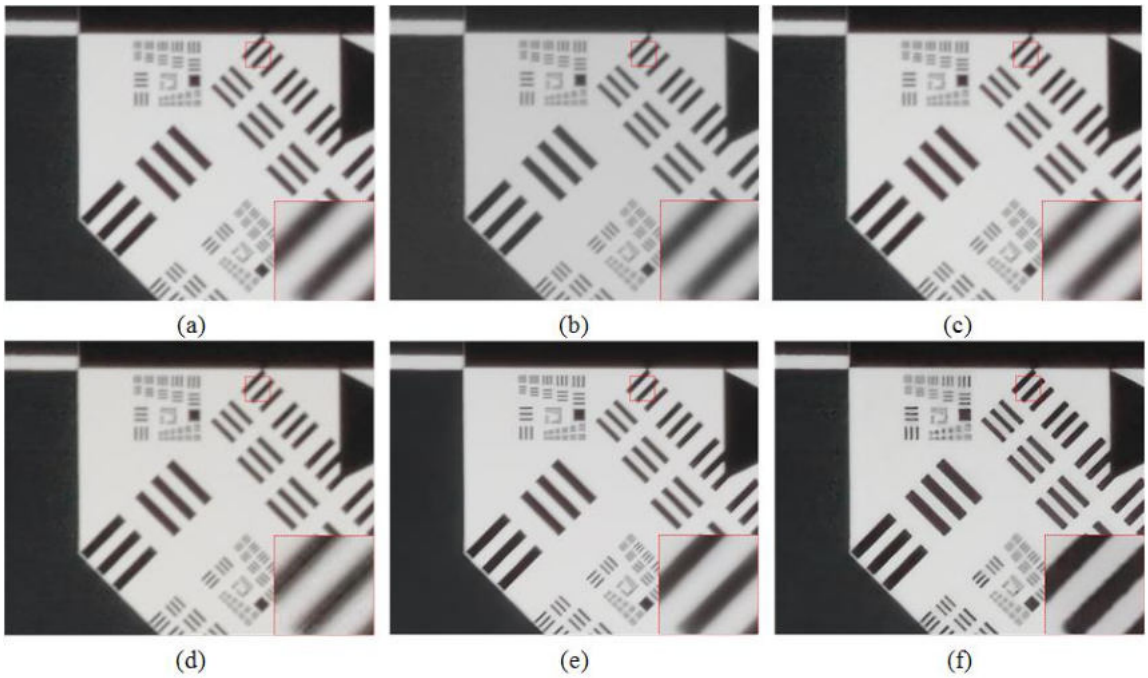


Figure 5.1: Real-word image super-resolution by different methods with scale factor 5; The image is cropped from camera resolution chart. The results are produced by: (a) Bicubic; (b) NCSR; (c) VDSR; (d) SRGAN; (e) SRMD; (f) Ours, respectively.

5.1 Introduction

Single image super-resolution (SISR) which aims to recover a high-resolution (HR) image from its degraded low-resolution (LR) image, has been receiving much attention. It can effectively overcome the resolution limitation of low-cost imaging sensors or enhance existing images. In general, a degraded LR image $\mathbf{y}$ can be formulated as:

$$\mathbf{y} = (\mathbf{k} \otimes \mathbf{x}) \downarrow_s + \mathbf{n}_s \tag{5.1}$$

where $\mathbf{k} \otimes \mathbf{x}$ denotes a convolution between a blur kernel $\mathbf{k}$ and a latent HR image $\mathbf{x}$, and the script $\downarrow_s$ is a subsequent downsampling operation with scale factor s, and $\mathbf{n}_s$ is additive white Gaussian noise (AWGN).

Since the SISR is a severely ill-posed inverse problem, prior knowledge is required to provide extra information for the esimation of HR image. Three cate-

gories of approaches have been proposed in the past decades, *i.e.*, interpolation-based, optimization-based, and learning-based methods. The interpolation methods (*e.g.*, nearest, bilinear and bicubic interpolations) construct new data points within the range of a discrete set of known data points. Interpolation methods are simple and efficient but suffer from severe edge and detail loss in the zoomed HR image. The optimization-based approaches explicitly model prior knowledge to estimate the HR image. They are flexible in incorporating versatile priors (*e.g.*, sparsity [123, 31] and non-local similarity [13]) tailored for SISR. However, these optimization-based methods often suffer from the high computational cost and complex parameter adjustment. Benefiting from joint optimization and end-to-end training, CNN has achieved unprecedented success in SISR. The learning-based approaches implicitly use prior knowledge by learning a direct mapping from external training data.

Although these finely designed modes can obtain high signal-to-noise ratio (PSNR) and visual quality in testing images downsampled by bicubic approach, they do not work well in practical applications, where an LR image is amplified directly without pre-bicubic downsampling as shown in Fig. 5.1. These compared methods used in Fig. 5.1 are quite representative. Bicubic is a classic baseline super-resolution method and NCSR [13] fully exploits the non-local self-similarity and sparse property of naturals images. VDSR [48] was the first to use the residual network, making great progress in the field of SISR. SRGAN aims to produce naturally looking images by introducing adversative learning. SRMD [129] considers the factor of blur kernel and noise level are considered in the construction of SRMD dataset. However, the results whether by solving sparse model or training simulated data suffer from detail and texture loss, and large edge blur in real-world super-resolution.

It is important to preserve edge and texture due to the high resolution of images captured by current digital or mobile devices, which is a big challenge for real-world image super-resolution. The key issue of SISR is how to establish the relationship

between the LR and HR image pair to be used for CNN training. An LR image is broadly considered as the degraded result by bicubic downsampling of an HR image in most deep learning based methods. The downsampling was manipulated on the HR image blurred by a Gaussian filter [107] instead of the original HR image. Multiple degradation was managed with via a dimensionality stretching strategy in [129]. However, single and multiple degradation cannot reflect the real degradation. To better model the real degradation process, we establish a new dataset captured by real cameras for discriminate learning without any assumption of blur kernel, noise and downsample operator.

The main contributions of this chapter are listed in the following: 1) A novel long-short focus dataset via digital cameras with zoom lens is developed to address the issue of real-world image super-resolution. 2) The effectiveness of this dataset is demonstrated by a plain CNN network, which exhibits evident advantage over simulated data. 3) According to the characteristics of the dataset, we employ a hybrid loss and a reversible downsampled operation in training.

5.2 Related work

Two key factors concerning image super-resolution are: dataset and network. Intensive work has been done to design complex network architecture and efficient loss functions, while few work has been reported about building new datasets. The pioneer work using CNN for SISR was proposed by Dong *et al.* [11] to learn a simple three-layer CNN (SRCNN). They extended their work in [12] by adding a deconvolutional layer and adopting smaller kernels in a deep layer network (FSRCNN). To overcome the vanishing-gradient problem in training deep CNN, residual learning and skip connection are commonly employed tricks in SISR. Kim *et al.* [48] utilised the residual learning strategy [37] to overcome the difficulty of training deep net-

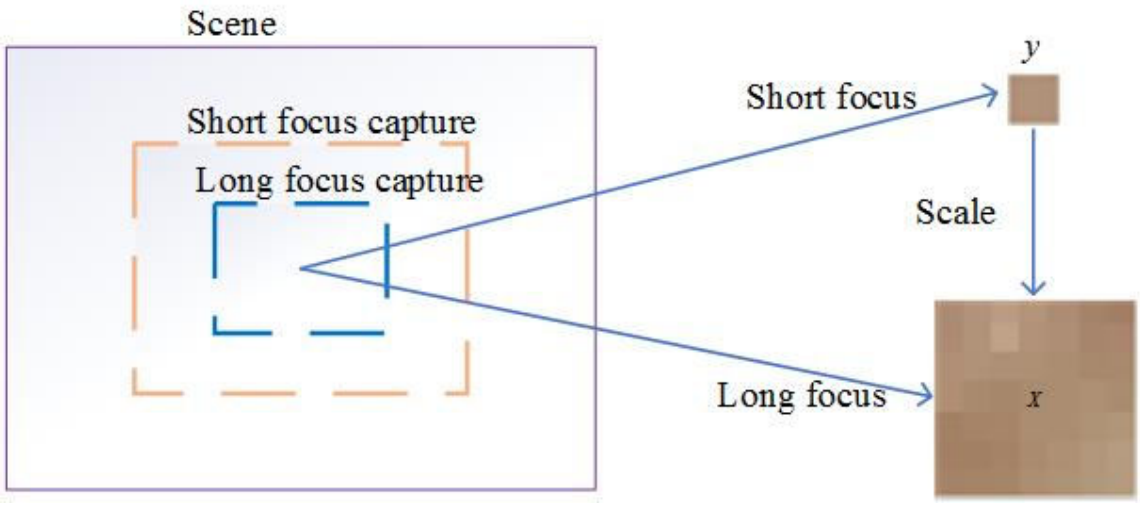


Figure 5.2: Illustration of rationality of database construction for real-image super-resolution.

works (VDSR). Residual deep learning brings positive effect on SISR as LR and HR images share similar low-frequency information. They got similar experimental outcomes by adding a recursive layer in [49] (DRCN). Tai *et al.* [113] achieved a very deep (52 convolutional layer) model, named DRRN, by combining global and local residual learning with recursive learning. Lai *et al.* [56] also utilised residual learning in a Laplacian pyramid framework (LapSRN) with a Charbonnier loss function. Dense convolutional network (Densenet) [40] was also applied to SISR in [114] (SR-DenseNet). The results generated by these methods can obtain high objective scores, but the results are not perceptually pleasing. To preserve more high-frequency information in HR images, generative adversarial network (GAN) was adopted in [59] (SRGAN) with both perceptual and adversarial loss. MSE loss measure was formulated on the feature extracted by VGG in [44]. Channel attention [134] was proposed to better rescale channel-wise features.

Some efforts have been made to improve the performance of image restoration or enhancement in the real scenario by building new datasets driven by physical imaging mechanism. For example, novel datasets have been built with regard to

denoising [93], deblur [88], and low-light image enhancement [8], obtaining better performance in the practical application. They all look into the traditional problems in a real-world manner. In denoising [93], the authors established the high and low ISO image pair for real photographs denoising. The images captured by low and high ISO were regarded as reference image and noise image, respectively. The noise image captured by high ISO can better reflect realistic noise, resulting in better denoising outcomes in real-world image denosing. By contrast, previous works emulate the input noise data by adding AWGN [127]. In deblur [88], they captured high-frame rate video using GOPRO4 HERO black camera, and got the blur image by averaging adjacent frames whose middle frame was referred to as the "ground-truth". Chen *et al.* [8] simultaneously denoised and enhanced the low exposure image using long exposure image as reference. Similarly, the authors in [41] built a dataset via mobile phones and high-end cameras for photo quality enhancement to enhance the images taken by mobile phones. However, few works has been reported to establish a dataset for the real-world image super-resolution.

In this chapter, we present a novel approach to obtain a new dataset by adjusting the lens focus of digital cameras, given that data plays a crucial role in data-driven approaches based on deep learning for SISR.

5.3 Dataset

For a scene as shown in Fig. 5.2, we can capture it with different filed of view (FOV) by adjusting lens focus. For one pixel y in the short-focus image, the long focus image with small FOV corresponds to a region X. In the super-resolution, the aim is to solve $x_i \in X$ from y, which is severely ill-posed. CNN based methods aim to learn the relationship between y and x_i by training large amounts of data. In the conventional data construction, it is widely implemented by bicubic operator. It is

difficult to model the process by training inverse single degradation. Therefore, we propose to establish real dataset to model more potential degradation types.

We use three representative digital cameras to build the dataset of over 300 long-short focus image pairs including Canon 600D (len range: 18mm-135mm) 190 pairs at resolution 3000×3000; Sony $\alpha 7$ (len range: 28mm-70mm) 204 pairs at resolution 500×500 ; Nikon D7000 (len range: 17mm-50mm) 214 pairs at resolution 500×500. The scale factor is within a range rather than an accurate value. Empirically, the scale factor is: Canon ($\times 4 - \times 6$), Sony ($\times 2 - \times 3$) and Nikon ($\times 2 - \times 3$). The process of dataset construction consists of two steps: long-short focus image capture, and HR-LR image pair crop and registration.

Long-short focus image capture: We capture the long-short focus image as shown in Fig. 5.3 (a) and (b), where the short focus image and long focus image are taken by turning the focal length of lens down and up, respectively. The images are taken mostly in outdoor scene, since the illumination fluctuation caused by incandescent lights may change drastically and largely in indoor scene. The indoor capture suffers from obvious luminance and color change between long and short image, as well as stripe effect when the shutter is faster than the flicker frequency of incandescent lights. The white balance is locked to "Daylight" or "Cloudy" to reduce the color variation according to the weather and conditions. The continuous auto-focus and partial metering are adopted during each shot. The tripod is used to avoid large pixel inconsistency. The aperture, ISO and shutter are adjusted manually during each shot to ensure that the scene center is exposed adequately. The principle is that ISO cannot be too large to avoid noise. The shutter speed should be rapid to avoid blur. Preferably, we choose a large aperture to capture a closer shot. Eventually, we delete some undesirable images such as over-exposure and under-exposure, and out of focus images.

HR-LR image crop and registration: After acquiring images of long and

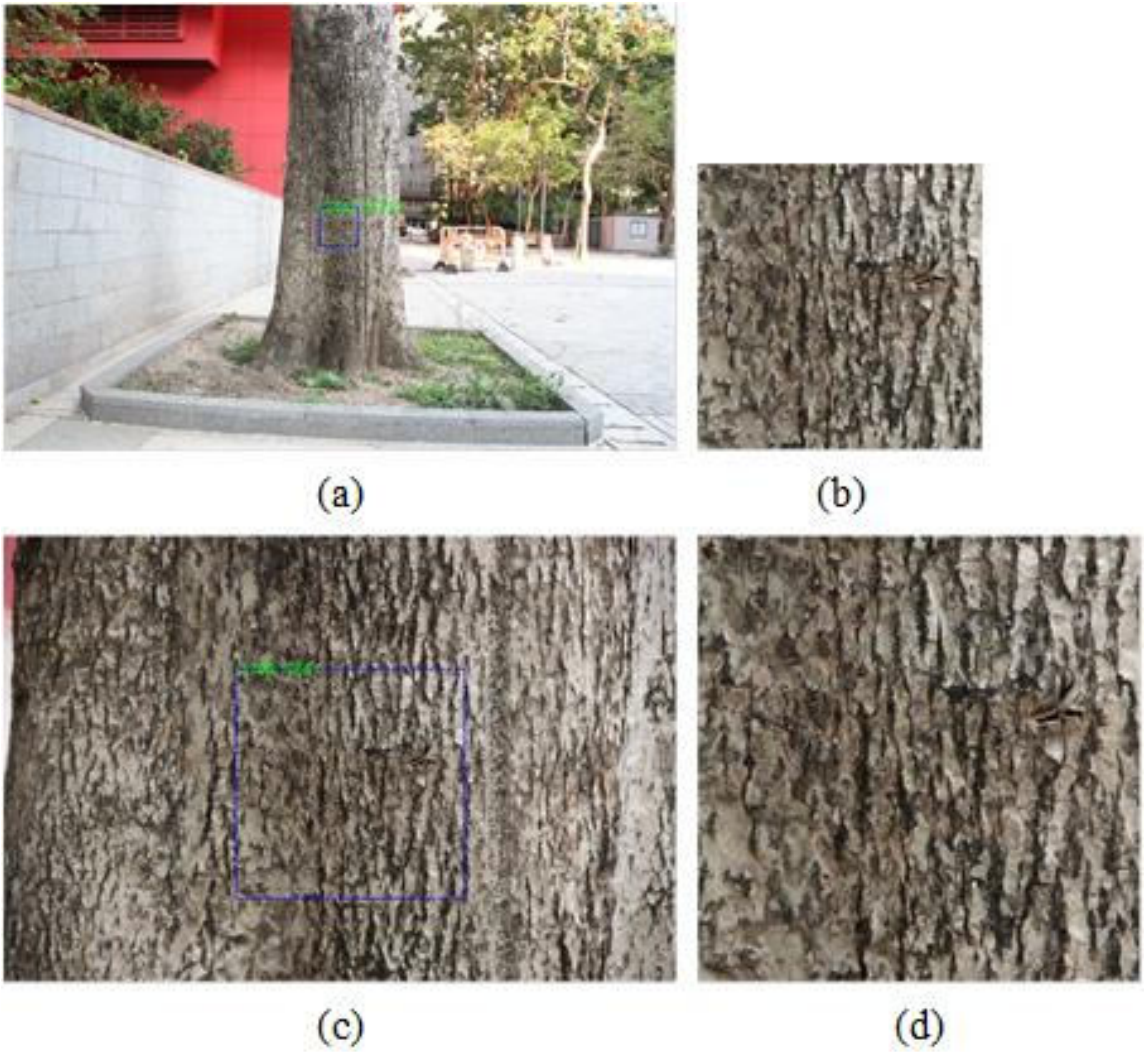


Figure 5.3: One example of image crop and registration, respectively; (a) is the short focus image (Canon 18mm); (b) is long focus image (Canon 135mm); (c) is the extracted LR image; (d) is extracted HR image.

short focus, the image crop and registration need to be done to get corresponding LR and HR image pair. To get a clear image without being affected by lens distortion, we only use pixels of center region of each image. First, we crop a central region X form the long-focus image, then we try to find the corresponding low-resolution patch Y from the short-focus image via exhaustive search. The distance or similarity

metrics can be expressed as:

$$Distance(\Upsilon(Y \uparrow s) - X) \tag{5.2}$$

where s means the bicubic upscale. s is varying pair by pair due to unavoidable geometric distortion of imaging, but it was found empirically it is within a certain range. The magnification power (upscale) is initialized empirically and fine-tuned via exhaustive search. Υ is a translation transform with horizontal and vertical offset. *Distance* is the judging criteria. Due to inevitable luminance change in short-long image pair, we employ FSIM [132] which is luminance-invariant as the similarity measure instated of mean squared error (MSE).

One example is given in Fig. 5.3, where we crop a central region with size of 150×150 in long-focus image as shown in Fig. 5.3 (c). The upscale is initialized as 5.6 by referring to size of the tree bark in an interactive interface displaying short-long focus image pair. We narrow the upscale range into 5.5-5.9, and search the similar region in a local neighbourhood of short-focus image as shown in Fig. 5.3 (a). The step of s is 0.1. After the fine-tuning process, the horizontal and vertical offset is determined as 88 and 210 deviated from the central point and the upscale is set as 5.8. Although we can obtain decent HR-LR image pairs via above search, we only contain the simple translation and scale factors in Equ. 5.2. Considering more complex non-rigid factors, the image registration is made to attain the eventual LR and HR image pair. Here, we use corrected scale-invariant feature transform (SIFT) descriptor [75] as the registration technique in view of the registration speed and robustness. The final extracted HR-LR pair can be seen in Fig. 5.3 (b) and (d).

To more conveniently compare with other super-resolution methods, we also provide dataset with stable upscale by stabilising s. We have got $\times 2$, $\times 2.5$, $\times 3$, and $\times 2 - \times 3$, $\times 4 - \times 6$ dataset. The scale augmentation is not suitable for our dataset, so we train each scale separately. The scale provided by our dataset is limited due

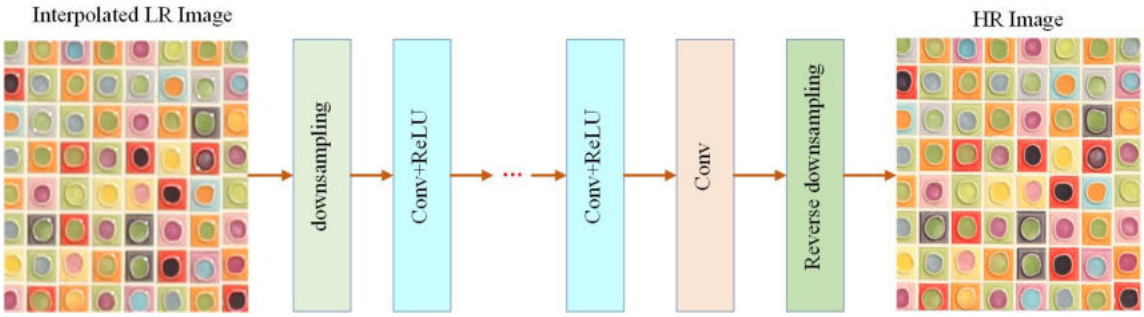


Figure 5.4: The architecture of the proposed network for real-world image super-resolution

to the difficulty in building database, but we can also obtain super-resolution results of other scales with the help of bicubic. If we want to have a $\times 8$ sup-resolution, we can input the LR image into the $\times 5$ network, and then interpolate the $\times 5$ result to the $\times 8$ HR image.

5.4 The proposed network

In this section, we validate the effectiveness of our datatset via deep learning which has been widely investigated in SISR. We train an end-to-end CNN to model the mapping relationship F between the HR and LR image pairs. The architecture of our designed network is depicted in Fig. 5.4.

5.4.1 Network overview

To show the effectiveness of the new built dataset, we resort to a simple CNN without complex network structure, where four types of operations are adopted: downsampling, Convolution (Conv) and Rectified Linear Units (ReLU), and reverse downsampling. The specific combination can be found in Fig. 5.4. The first layer is a reversible sub-pixel operator[107, 128] which divides an LR image with size $W \times H \times C$ into four downsampled sub-images with size $\frac{W}{2} \times \frac{H}{2} \times C$. C is the channel of the input image. Similarly, the last layer is the reverse operation to reshape the processed sub-images

into the final HR image. The convolution part of the network has D layers. The size of filter is $f1 \times \ f1 \times c \times n$. $f1 \times f1$ is the kernel size, and c and n is the number of channel and filter, respectively. Batch normalization (BN) is not used because we found empirically that it cannot bring evident gain in super-resolution.

It is indispensable to perform upscaling to enlarge the resolution of the LR image into HR space at some point. Upscaling can be done by prior bicubic upsampling or learnt within CNN. Learning upscaling via a deconvolution layer helps reduce GPU memory cost and improve the reconstruction accuracy. However, we have to handle the bicubic upscaling before the LR image is fed to the network owing to the unfixed scale of our dataset. To reduce the computational complexity resulting from the feature extraction via nonlinear convolutions in HR space, we divide the interpolated LR image into several sub-images via a reversible downsampling operator. Furthermore, downsampling operation also expands receptive field without increasing the network depth which in turn lead to moderate number of parameters. Our dataset suffers from slight pixel inconsistency although registration has been done in the preprocessing of dataset. It is hard to address this issue by devising loss functions, as most loss functions are pixel sensitive. Some distribution consistency metrics could be considered, but these metrics are usually not convex or differentiable. We handle the pixel inconsistency by adopting a bigger receptive field. We found through experiments that a bigger receptive field is less sensitive to slight non-registration, leading to more robust performance. Another strategy to obtain large receptive field is to use dilated filter convolution. However, dilated filter is prone to result in artifacts around sharp edges.

5.4.2 Loss functions

We exploit the performance of our dataset under several representative loss functions and design two hybrid loss functions. First, different loss functions are tried including

MSE loss, L_1 norm loss and perceptual loss (SSIM, multi-scale SSIM). Given a training set $\{I^i_{LR}, I^i_{HR}\}^N_{i=1}$, the pixel-wise MSE loss can be formulated as:

$$L^{l_2}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|F(I^i_{LR}) - I^i_{HR}\|^2_F \tag{5.3}$$

where Θ denotes the parameter of the CNN network. Due to the fast convergence properties, MSE measure has been widely utilized in regression problems. But it is not consistent with human visual system (HSV), which is easy to lead to over-smoothing results. We consider other loss functions in our training. The L_1 loss can be calculated as:

$$L^{l_1}(\Theta) = \frac{1}{N}\sum_{i=1}^{N} \|F(I^i_{LR}) - I^i_{HR}\|_1 \tag{5.4}$$

Compared with MSE, L_1 loss can generate more visually friendly results, but some artefact also appeared. The perceptual SSIM loss can be calculated as:

$$L^{SSIM}(\Theta) = \frac{1}{N}\sum_{i=1}^{N}(1 - ssim(F(I^i_{LR}) - I^i_{HR})) \tag{5.5}$$

SSIM loss can well reflect the local structure information. The multi-scale SSIM (MSSIM) was also used. Besides, we combine MSE and L_1 loss, MSE and SSIM loss to take advantage of different loss properties. The results by different loss functions can be found in Fig. 5.5 (a-f).

5.5 Experimental results

In this section, we evaluate the performance of our method on our test dataset and general non-reference image super-resolution.

5.5.1 Implementation detail

The initiation of learning rate is set to 0.1 and then decreases by a factor of 10 when the training error stops decreasing for 10 epochs. The batch size is set to 128. The dilation is set to 1 with zero-padding of 1. The layer depth D is 15 and filter size $f1$ is 3. We use 64 and 128 filters of the size 3×3 for shallow and deep weight layers, respectively. The filter number n is 1 in the last layer for image reconstruction. Data augmentation with geometric transformations techniques (horizontal flip and rotation with 3 directions) is implemented to prevent our network from overfitting. The images in the training set are split into 41 by 41 or 96 by 96 patches with the stride of 41 or 96 and mini-batch size is set 64 for stochastic gradient descent (SGD). In the training phase, we use patches of $41 \times 41 \times 279360$ and $96 \times 96 \times 207616$ for scale 2-3 and scale 4.5-6, respectively. C is set to 1 to reduce the impact of chrominance variation. We only use the luminance data in YCbCr color space instead of RGB channels in training. The LR input is interpolated by bicubic into the same resolution with HR image. We record both the RAW and JPEG images. For simplicity, we only use the JPEG images to avoid the complex camera pipeline. We implement our model with Matcovnet and Pytorch libraries. All the experiments were performed in a desktop with i7-4790K CPU and NVIDIA 1080 GPU.

5.5.2 Results with different loss functions

In this section, we self-evaluate our dataset via different loss functions on our benchmark datasets as shown in Fig. 5.5. One can see that the MSE can produce good results, but the results are a little bit over-smoothing. The l_1 norm loss brought sharper edge, but some artifacts also occurred. Compared with SSIM, the results by MSSIM preserve more local structures. The results produced by mixed loss function $0.95 \times l_1 + 0.05\times$ MSE have weighted performance of l_1 norm and MSE. $0.95\times$ MSE

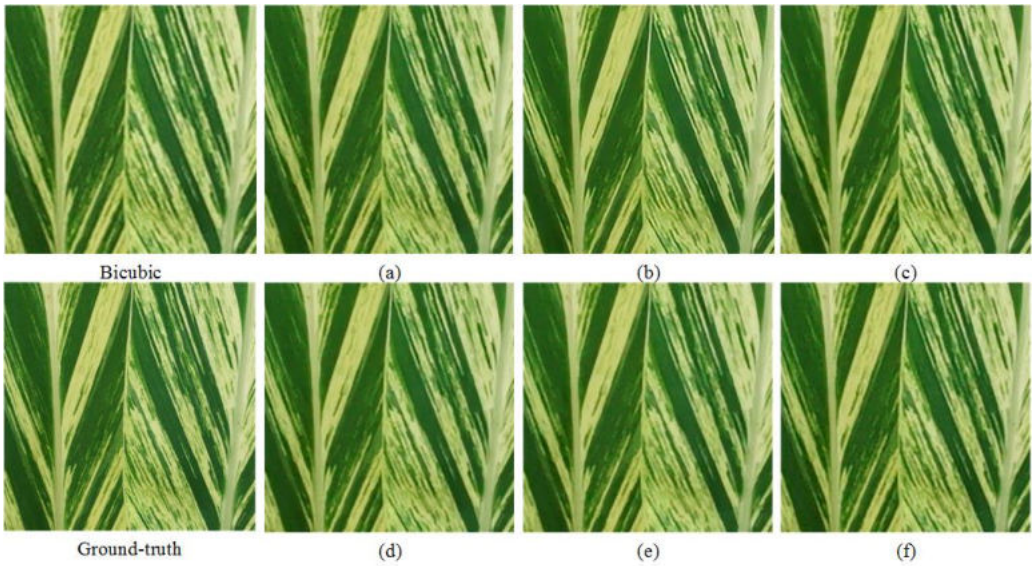


Figure 5.5: The results by different loss functions on our dataset (upscale=2); (a) MSE; (b) l_1 norm; (c) SSIM; (d) MSSIM; (e) l_1 norm + MSE; (f) MSE+SSIM

+ 0.05× SSIM have weighted performance of MSE and SSIM. The result by the combined l_1 and MSE loss shows less artifacts than l_1 and SSIM, and better edge preservation than MSE. We adopt this loss scheme in the subsequent comparison.

5.5.3 Experiments on our test dataset

In this section, first we give the experimental results on our test dataset as shown in Fig. 5.6 with scale 2 and Fig. 5.7 with scale 5. In Fig. 5.6, we compare our method with 6 the-state-of-art methods. Our method exhibits best visual quality and least artifacts comparable with ground-truth. The results by compared methods expect SRMD present similar performance of serve edge and detail loss. It indicates that models trained by bicubic down-sampled dataset cannot super-resolve real-world images without prior degradation. SRMD adds blur kernel and noise level in creating training dataset, obtaining better visual quality than other models. But it is still not applicable for the real super-resolution, since estimating blur kernel in real image super-resolution is difficult as shown in Fig. 5.6 where inaccurate kernel width

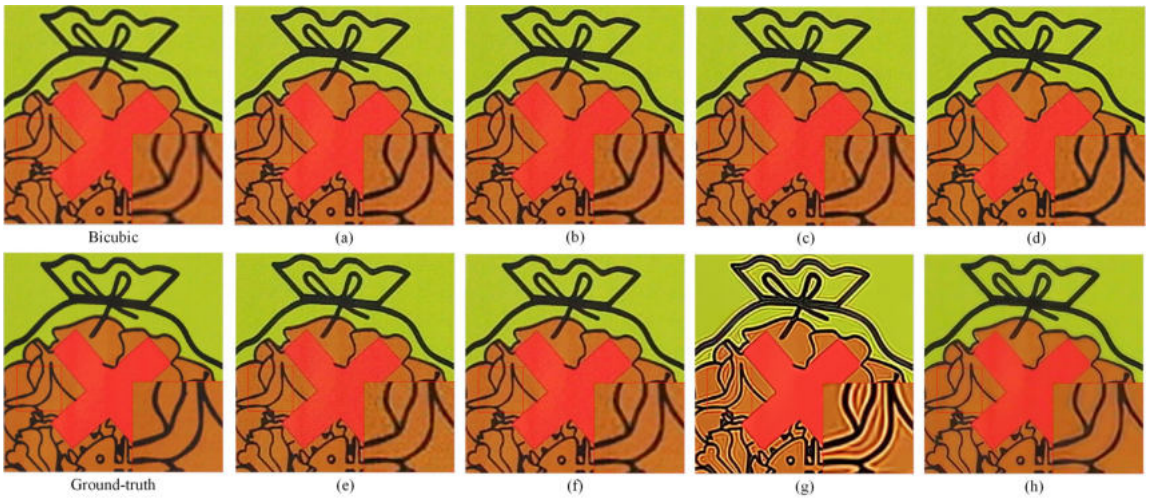


Figure 5.6: The results by different methods in one test image (upscale=2); (a) SRCNN; (b) VDSR; (c) DRNN; (d) LapSRN; (e) SRGAN; (f) Waifu2x; (g) SRMD (h) Ours

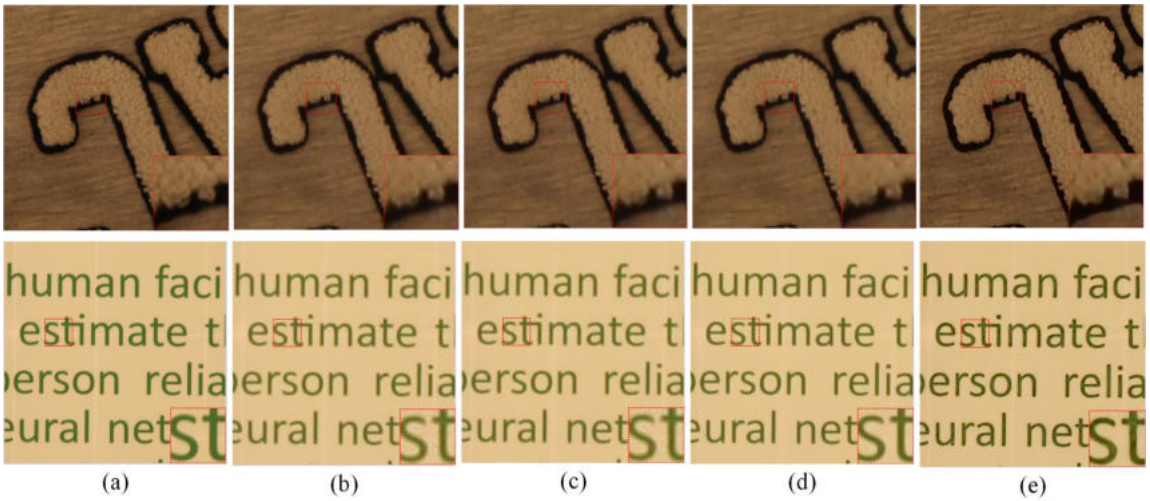


Figure 5.7: Test results by different methods in one test image (upscale=5); (a), (b), (c), (d) and (e) are the results by ground-truth, bicubic, VDSR, SRMD and ours, respectively.

Table 5.1: The PSNR scores by different methods on our test dataset (upscale=2)

Methods	SRCNN	VDSR	DRNN	LapSR	SRGAN	SRMD	Waifu2x	MSE	L1	SSIM	L1+MSE
Metrics	PSNR										
Image1	24.55	24.87	24.74	24.97	24.00	17.47	24.76	26.57	25.92	26.70	25.25
Image2	27.08	27.15	27.17	27.57	26.82	14.24	27.33	28.27	26.85	27.25	27.43
Image3	25.36	25.58	25.56	27.93	25.10	17.26	25.47	26.63	25.93	26.68	26.27
Image4	23.85	23.77	23.64	24.03	23.29	17.37	24.29	26.90	24.38	25.73	24.72
Image5	25.01	25.21	25.22	25.76	24.96	14.53	25.18	25.54	25.02	25.53	25.09
Image6	17.66	18.02	18.02	17.05	16.97	11.63	16.81	19.07	18.66	19.05	19.07
Image7	20.90	21.48	21.48	20.14	19.95	14.26	19.76	24.54	22.53	23.98	24.17
Image8	24.68	25.22	25.18	24.20	23.56	19.14	23.52	26.11	25.67	26.15	25.74
Average	23.64	23.91	23.87	23.96	23.08	15.74	23.39	25.45	24.37	25.13	24.72

induced severe visual artifact.

In Fig. 5.7, we use less comparing methods, since scale 5 is not available in the released pre-trained models. We have to re-train the network to obtain scale 5 models. For saving time, we only train two representative VDSR and SRMD with the default parameters which were optimized by the authors. It is enough because from Fig. 5.6, we can know that the results by other methods are similar with VDSR in real super-resolution. Our results can better recover shaper edge information than the compared methods. For instance, the branch and letter labelled in red square box of our method have clearer edges than compared methods.

Furthermore, we use PSNR and structural similarity (SSIM) [117] indexes to evaluate the methods quantitatively as shown in Table. 5.1 and Table. 5.2. It is difficult to make quantitative comparisons with other methods, since the scale of our dataset is unfixed. To make fair comparison, we only make the objective experiments on the $\times 2$ dataset. It also should be noted that the ground-truth of our dataset has the problem of local focal de-focus owning to the change of depth of field. But small numbers of negative samples are acceptable. Our implementations with different loss functions all have higher PSNR and SSIM values than the compared methods.

Table 5.2: The SSIM scores by different methods on our test dataset (upscale=2)

Methods	SRCNN	VDSR	DRNN	LapSR	SRGAN	SRMD	Waifu2x	MSE	L1	SSIM	L1+MSE
Metrics	SSIM										
Image1	0.86	0.87	0.86	0.87	0.84	0.54	0.88	0.92	0.91	0.94	0.91
Image2	0.81	0.81	0.81	0.83	0.79	0.35	0.82	0.84	0.80	0.84	0.82
Image3	0.77	0.77	0.77	0.83	0.73	0.58	0.78	0.83	0.82	0.84	0.82
Image4	0.87	0.87	0.87	0.89	0.83	0.70	0.89	0.92	0.90	0.92	0.91
Image5	0.79	0.79	0.79	0.81	0.77	0.45	0.80	0.79	0.78	0.80	0.78
Image6	0.63	0.64	0.64	0.54	0.53	0.37	0.53	0.64	0.62	0.68	0.60
Image7	0.77	0.77	0.77	0.67	0.66	0.39	0.67	0.79	0.76	0.79	0.78
Image8	0.86	0.87	0.87	0.87	0.84	0.77	0.85	0.88	0.88	0.88	0.87
Average	0.80	0.80	0.80	0.79	0.75	0.52	0.78	0.83	0.81	0.84	0.81

5.5.4 Experiments on general real-world images

To evaluate the generalization ability of our model, we test referenceless image datasets including Set291, Set5, BSD [81], McM [131], Super-chart, and cellphone images. Set291, Set5, BSD, McM are collected from internet. Super-chart is captured by Sony RX100, which is different from the camera we use in the dataset construction. Cellphone images are taken by ourselves with three mobile phones (Huawei Mate8, Google Pixel 1, Iphone 6). Some visual results can be found in Fig. 5.8. VDSR presents similar performance in real non-reference image super-resolution with bicubic upsampling. The results suffer from obvious texture and edge loss. SRMD and our method provide better visual qualities, effectively enhancing edges and providing more details. The results by SRMD are unreal with over-enhanced edge and over-smoothing details which can be seen from the stipe, mushroom roof and plant leaf labelled in red boxes. Compared with SRMD, our results appear more natural with faithful structure and texture preservation. The models trained by bicubic degradation fail to work in real situation. SRMD with multiple degradation faces the problem of parameter choice of kernel-width, although it can bring obvious gains than other models. We use the kernel-width of 1.7 and 3 for scale 2 and scale 5, respectively. Our method is free of parameter adjustment in the test stage.

Some non-reference metrics including Niqe [86] and Brisque [85] are used to eval-

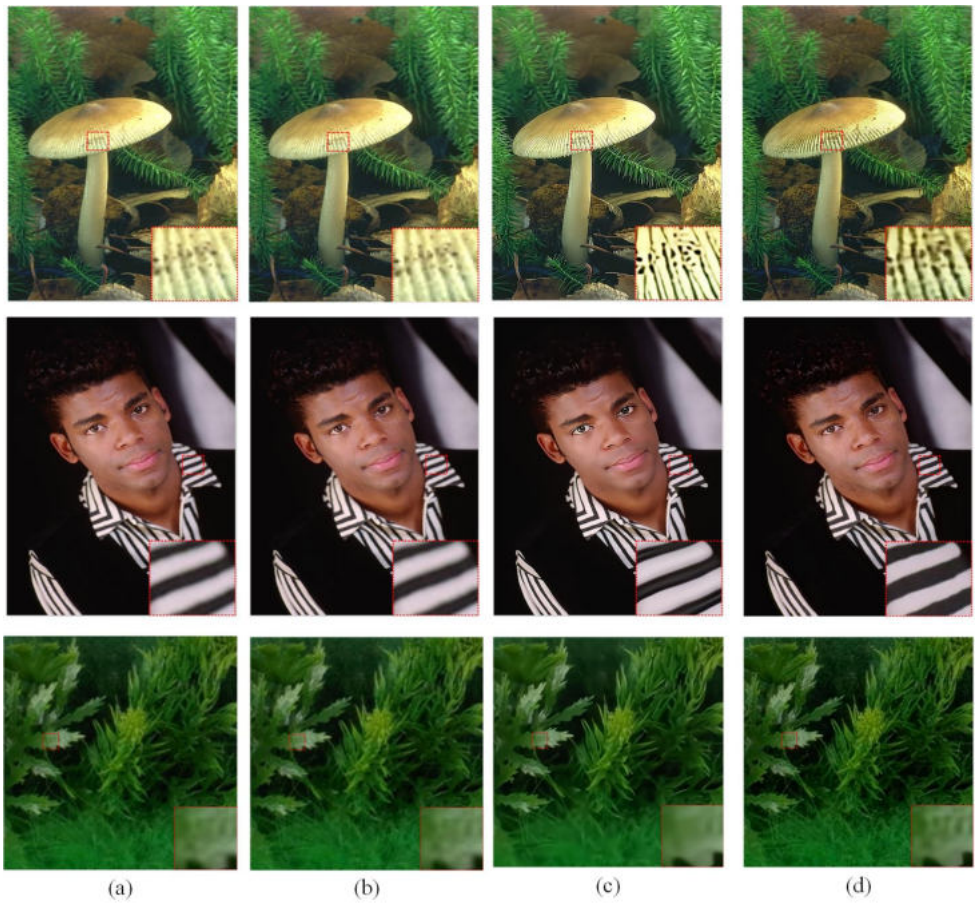


Figure 5.8: Test results in general image super-resolution (upscale=5); (a), (b), (c) and (d) are the results by bicubic, VDSR, SRMD and ours, respectively.

Table 5.3: The non-reference objective scores of different methods on general real-world image super-resolution (upscale=5)

Models	Bicubic		VDSR		SRMD		Ours	
Dataset	[86]	[85]	[86]	[85]	[86]	[85]	[86]	[85]
Set291	9.66	65.92	8.67	60.68	8.74	79.25	**8.28**	**54.15**
BSD	9.09	67.93	8.10	62.89	**7.20**	61.14	7.55	**56.92**
Cellphone	10.36	69.41	9.38	64.44	9.82	77.32	**9.01**	**60.48**
Sup-chart	11.09	73.32	10.14	70.98	11.80	84.16	**8.70**	**57.76**
McM	9.34	67.53	8.25	64.43	**7.03**	66.26	7.60	**56.37**
Set5	9.55	69.63	8.45	62.18	7.73	65.23	**7.36**	**54.08**
Average	9.85	68.96	8.83	64.27	8.72	72.23	**8.08**	**56.63**

uate the results without ground truth in Table 5.3. The non-reference scores are consistent with visual qualities. Niqe and Brisque enphasize the naturalness and spatial quality. The smaller values reflect better perceptual quality. The results by our method have highest mean Brisque and Niqe values among the four representative methods, reflecting that the results by our method are visually plausible in terms of naturalness and spatial quality. Bicubic and VDSR suffer from large edge blur and the loss of details, leading to low Brisque and Niqe values. SRMD preserves sufficient sharp information at the cost of destroyed spatial textures in terms of lowest Brisque values. In summary, our method strikes good balance among spatial quality, naturalness, edge and texture preservation.

5.6 Conclusion

In this chapter, we made the first attempt to address the issue of real-world image super-resolution by establishing a long-short focus image dataset, which contains real image pairs of low and high resolutions. Such a dataset taken by real cameras can describe better the real degradation of low resolution images than the simple bicubic kernel, which are commonly used to simulate data for super-resolution network training. A fast plain network with hybrid L_1 norm and MSE loss was deployed on our dataset. We adopted the reversible downsampling scheme to use bigger receptive field to address the non-perfect registration issue in the training stage. The experiments on our dataset and other real images from six datasets demonstrated that the super-resolution network trained our dataset achieves remarkable results compared with previous state-of-the-art methods trained on simulated data. It shows much better visual qualities with good edge and texture preservation.

Chapter 6

Conclusions and Future Work

6.1 Conclusions:

In this thesis, we focus on the key techniques about HDRI and super-resolution. We propose new methods for tone mapping, multi-exposure image fusion, and real image super-resolution. The contributions by the thesis can be summarized as:

(1) We propose a clustering based locally adaptive tone mapping operator. Each patch is decomposed into three components. The low frequency is compressed and high frequency is enhanced. The structure component is adaptively projected on a dictionary containing similar structures which can sufficiently express the local structure. The tone mapping process is implemented on the projection coefficients instead of the original intensity, which is easier to adjust the details. We simultaneously process the luminance and color information so as to preserve more color information. The multi-scale version is presented via progressively decomposing the patch images, which can effectively decrease the halo effect. An off-line version is implemented via pre-training PCA transforms from natural images to reduce the computational cost. Experimental results on extensive synthetic and camera raw data demonstrate the effectiveness of the proposed method qualitatively and quantitatively.

(2) A multi-scale fast patch-decomposition multi-exposure method is developed. We revise the structure patch decomposition by deleting the normalization of fused structure component. The patch decomposition and aggregation with any size of stride step can be denoted as mean filtering, which largely reduce the computational cost. The multi-scale SPD-MEF is realized by progressively decomposing the mean image formed by the means of all patches to reduce the halo effect. Different weight function is compared in fusing the low frequency. Extensive experimental results indicate that it can produce pleasing fusion results with less artifacts and reduced computational cost in both static and dynamic scenes.

(3) We explore the application of deep learning in MEF. Given that the feature is important for determining the weight for fusion. First, we use the features extracted via a pre-trained convolutional network. For each pixel, we can obtain a dense feature vector whose dimension is the number of feature maps. Different sorts of networks are compared for feature extraction. With the feature map extracted from convolutional layer, we compute the local contrast and consistency map for the weight map. The local contrast is defined by the L_1 norm of the feature vector. The consistency map is computed by the Euclidean metric of corresponding feature vectors for motion detection in dynamic situation. The proposed method can work for both static and dynamic scenes. Moreover, we explore the implementation of end-to-end MEF network.

(4) We address the real image super-resolution by capturing real camera lens data. We establish a novel long-short focus dataset by adjusting digital cameras with zoom lens. The LR and HR image pairs are cropped via image registration. A plain CNN network a reversible downsampled operation in training is employed considering the unfixed scales. The trained network with the built dataset

can bring more fine details compared with traditional simulated data driven network especially in real-world image super-resolution.

6.2 Future Work:

(1) The current approach assumes the input sequence is perfectly aligned. To make it more practical, it would be interesting to investigate how the structural consistency measurement in Eq. (3.14) can be used for robust camera motion alignment [19]. Like most existing deghosting schemes, our method may fail in certain extreme cases. For example, if under-/over-exposed regions contain moving objects, the binary maps for region segmentation would be less accurate, and visible ghosting artifacts may appear in the final image. Therefore, it is desirable to make better use of the binary maps at different scales, *i.e.*, set proper scale-dependent thresholds in Eq. (3.15) and integrate these maps for improved object motion detection.

(2) Another interesting direction to explore is how exposure bracketing is practiced to capture an optimal set of input images for a given MEF algorithm in either radiance or intensity domain. Hasinoff *et al.* [34] defined the optimality in terms of worse case signal-to-noise ratio, and they found that much higher and variable ISO settings lead to better noise reduction in darkest regions. Gupta *et al.* [33] defined the optimality in terms of image registration, and found that a Fibonacci bracketing strategy, where each exposure time is the sum of the previous exposures, better serves the purpose. Later, Hasinoff *et al.* [35] gave up exposure bracketing and deliberately captured images of constant under-exposes, which essentially transfers HDR imaging to a burst denoising problem. All the three strategies target at different points along the HDR image pipeline. A more desirable solution to exposure bracketing would

be optimized for the perceptual quality of fused images, which will ultimately be consumed by our visual systems. Although we have observed substantial progress of developing MEF algorithms, computational models that can automatically assess the perceptual quality of fused images are largely lacking, especially in the case of dynamic scenes. Since objective quality models form a cornerstone in image processing and computational photography, such a model for MEF of dynamic scenes would immediately lead to fair algorithm comparison and better algorithm design.

(3) We separately study two image enhancement tasks: HDRI and super-resolution. An interesting future work is to combine HDRI and super-resolution. It is really practical to simultaneously enhance the dynamic range and resolution of an image. A direct idea is to generate pseudo a multi-exposure sequence via the gamma transform of LR image. To this end, a deep network can be trained to enhance both the dynamic range and resolution.

Bibliography

[1] Codruta O Ancuti, Cosmin Ancuti, Christophe De Vleeschouwer, and Alan C Bovik. Single-scale fusion: An effective approach to merging images. *IEEE Transactions on Image Processing*, 26(1):65–78, Jan. 2017.

[2] Alessandro Artusi, Thomas Richter, Touradj Ebrahimi, and Rafal K Mantiuk. High dynamic range imaging technology [lecture notes]. *IEEE Signal Processing Magazine*, 34(5):165–172, 2017.

[3] Tunç Ozan Aydin, Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Dynamic range independent image quality assessment. *ACM Transactions on Graphics*, 27(3):69, 2008.

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

[5] Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced high dynamic range imaging: theory and practice*. CRC Press, 2011.

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

[7] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 2018.

[8] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. *arXiv preprint arXiv:1805.01934*, 2018.

[9] Hwann-Tzong Chen, Tyng-Luh Liu, and Chiou-Shann Fuh. Tone reproduction: A perspective from luminance-driven perceptual grouping. *International Journal of Computer Vision*, 65(1-2):73–96, 2005.

[10] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Com-*

puter graphics and interactive techniques, pages 369–378. ACM Press/Addison-Wesley Publishing Co., 1997.

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199. Springer, 2014.

[12] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016.

[13] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.

[14] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

[15] Frédéric Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. In *Computer Graphics Forum*, volume 22, pages 419–426. Wiley Online Library, 2003.

[16] 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.

[17] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *ACM Transactions on Graphics*, volume 21, pages 257–266. ACM, 2002.

[18] Ashley Eden, Matthew Uyttendaele, and Richard Szeliski. Seamless image stitching of scenes with large motions and exposure differences. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2498–2505, 2006.

[19] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, Oct. 2008.

[20] Yuming Fang, Hanwei Zhu, Kede Ma, and Zhou Wang. Perceptual quality assessment of HDR deghosting algorithms. In *IEEE International Conference on Image Processing*, pages 3165–3169, 2017.

[21] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM Transactions on Graphics*, volume 27, page 67. ACM, 2008.

[22] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In *ACM Transactions on Graphics*, volume 21, pages 249–256. ACM, 2002.

[23] Sira Ferradans, Marcelo Bertalmio, Edoardo Provenzi, and Vincent Caselles. An analysis of visual adaptation and contrast perception for tone mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2002–2012, 2011.

[24] Orazio Gallo, Natasha Gelfandz, Wei-Chao Chen, Marius Tico, and Kari Pulli. Artifact-free high dynamic range imaging. In *IEEE International Conference on Computational Photography*, pages 1–7. IEEE, 2009.

[25] A Ardeshir Goshtasby. Fusion of multi-exposure images. *Image and Vision Computing*, 2005.

[26] Michael D Grossberg and Shree K Nayar. Determining the camera response from images: What is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, Nov. 2003.

[27] Bo Gu, Wujing Li, Jiangtao Wong, Minyun Zhu, and Minghui Wang. Gradient field multi-exposure images fusion for high dynamic range image visualization. *Journal of Visual Communication and Image Representation*, 2012.

[28] Bo Gu, Wujing Li, Minyun Zhu, and Minghui Wang. Local edge-preserving multiscale decomposition for high dynamic range image tone mapping. *IEEE Transactions on Image Processing*, 22(1):70–79, 2013.

[29] Huxiang Gu, Ying Wang, Shiming Xiang, Gaofeng Meng, and Chunhong Pan. Image guided tone mapping with locally nonlinear model. In *European Conference on Computer Vision*, pages 786–799. Springer, 2012.

[30] Ke Gu, Shiqi Wang, Guangtao Zhai, Siwei Ma, Xiaokang Yang, Weisi Lin, Wenjun Zhang, and Wen Gao. Blind quality assessment of tone-mapped images via analysis of information, naturalness, and structure. *IEEE Transactions on Multimedia*, 18(3):432–443, 2016.

[31] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1823–1831, 2015.

[32] Gabriele Guarnieri, Stefano Marsi, and Giovanni Ramponi. High dynamic range image display with halo and clipping prevention. *IEEE Transactions on Image Processing*, 20(5):1351–1362, 2011.

[33] Mohit Gupta, Daisuke Iso, and Shree K Nayar. Fibonacci exposure bracketing for high dynamic range imaging. In *IEEE International Conference on Computer Vision*, pages 1473–1480, 2013.

[34] Samuel W Hasinoff, Frédo Durand, and William T Freeman. Noise-optimal capture for high dynamic range photography. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 553–560, 2010.

[35] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics*, 35(6):192:1–192:12, Nov. 2016.

[36] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[38] Jun Hu, Orazio Gallo, and Kari Pulli. Exposure stacks of live scenes with hand-held cameras. In *Springer European Conference on Computer Vision*, pages 499–512. 2012.

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

[40] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[41] Andrey Ignatov, Nikolay Kobyshev, Kenneth Vanhoey, Radu Timofte, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *IEEE International Conference on Computer Vision*, 2017.

[42] 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, Mar. 2008.

[43] Daniel J Jobson, Zia-ur Rahman, and Glenn A Woodell. Properties and performance of a center/surround retinex. *IEEE transactions on Image Processing*, 6(3):451–462, 1997.

[44] 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.

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

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

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

[48] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016.

[49] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1637–1645, 2016.

[50] Seon Joo Kim and Marc Pollefeys. Robust radiometric calibration and vignetting correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):562–576, 2008.

[51] Fei Kou, Weihai Chen, Changyun Wen, and Zhengguo Li. Gradient domain guided image filtering. *IEEE Transactions on Image Processing*, 24(11):4528–4539, 2015.

[52] Fei Kou, Zhengguo Li, Changyun Wen, and Weihai Chen. Multi-scale exposure fusion via gradient domain guided image filtering. In *IEEE International Conference Multimedia and Expo*, pages 1105–1110. IEEE, 2017.

[53] Fei Kou, Zhengguo Li, Changyun Wen, and Weihai Chen. Edge-preserving smoothing pyramid based multi-scale exposure fusion. *Journal of Visual Communication and Image Representation*, 53:235–244, May 2018.

[54] Jiangtao Kuang, Garrett M Johnson, and Mark D Fairchild. icam06: A refined image appearance model for hdr image rendering. *Journal of Visual Communication and Image Representation*, 18(5):406–414, 2007.

[55] Debarati Kundu, Deepti Ghadiyaram, Alan Bovik, and Brian Evans. No-reference quality assessment of tone-mapped hdr pictures. *IEEE Transactions on Image Processing*, 2017.

[56] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate superresolution. In *IEEE*

Conference on Computer Vision and Pattern Recognition, volume 2, page 5, 2017.

[57] Valero Laparra, Alexander Berardino, Johannes Ballé, and Eero P. Simoncelli. Perceptually optimized image rendering. *Journal of the Optical Society of America A*, 34(9):1511–1525, Sep. 2017.

[58] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.

[59] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 4, 2017.

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

[61] Joon-Young Lee, Yuki Matsushita, Boxin Shi, In So Kweon, and Katsushi Ikeuchi. Radiometric calibration by rank minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):144–156, 2013.

[62] Hui Li, Xixi Jia, and Lei Zhang. Clustering based content and color adaptive tone mapping. *Computer Vision and Image Understanding*, 2017.

[63] Hui Li and Xiao-Jun Wu. Densefuse: A fusion approach to infrared and visible images. *IEEE Transactions on Image Processing*, 28(5):2614–2623, 2018.

[64] Shutao Li and Xudong Kang. Fast multi-exposure image fusion with median filter and recursive filter. *IEEE Transactions on Consumer Electronics*, 2012.

[65] Shutao Li, Xudong Kang, and Jianwen Hu. Image fusion with guided filtering. *IEEE Transactions on Image Processing*, 2013.

[66] Yuanzhen Li, Lavanya Sharan, and Edward H Adelson. Compressing and companding high dynamic range images with subband architectures. In *ACM Transactions on Graphics*, volume 24, pages 836–844. ACM, 2005.

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

[68] Zhengguo Li and Jinghong Zheng. Visual-salience-based tone mapping for high dynamic range images. *IEEE Transactions on Industrial Electronics*, 61(12):7076–7082, 2014.

[69] ZhengGuo Li, JingHong Zheng, and S. Rahardja. Detail-enhanced exposure fusion. *IEEE Transactions on Image Processing*, 2012.

[70] Zhengguo Li, Jinghong Zheng, Zijian Zhu, and Shiqian Wu. Selectively detail-enhanced fusion of differently exposed images with moving objects. *IEEE Transactions on Image Processing*, 23(10):4372–4382, Oct. 2014.

[71] Zhengguo Li, Jinghong Zheng, Zijian Zhu, and Shiqian Wu. Selectively detail-enhanced fusion of differently exposed images with moving objects. *IEEE Transactions on Image Processing*, 23(10):4372–4382, 2014.

[72] Zhengguo Li, Jinghong Zheng, Zijian Zhu, Wei Yao, and Shiqian Wu. Weighted guided image filtering. *IEEE Transactions on Image Processing*, 24(1):120–129, 2015.

[73] Yu Liu and Zengfu Wang. Dense sift for ghost-free multi-exposure fusion. *Journal of Visual Communication and Image Representation*, 31:208–224, 2015.

[74] Kede Ma, Zhengfang Duanmu, Hojatollah Yeganeh, and Zhou Wang. Multi-exposure image fusion by optimizing a structural similarity index. *IEEE Transactions on Computational Imaging*, 2017.

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

[76] Kede Ma and Zhou Wang. Multi-exposure image fusion: A patch-wise approach. In *IEEE International Conference on Image Processing*, pages 1717–1721. IEEE, 2015.

[77] Kede Ma, Hojatollah Yeganeh, Kai Zeng, and Zhou Wang. High dynamic range image compression by optimizing tone mapped image quality index. *IEEE Transactions on Image Processing*, 24(10):3086–3097, 2015.

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

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

[80] Rafał Mantiuk, Scott Daly, and Louis Kerofsky. Display adaptive tone mapping. In *ACM Transactions on Graphics*, volume 27, page 68. ACM, 2008.

[81] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision*, volume 2, pages 416–423. IEEE, 2001.

[82] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion: A simple and practical alternative to high dynamic range photography. *Computer Graphics Forum*, 2009.

[83] Laurence Meylan and Sabine Susstrunk. High dynamic range image rendering with a retinex-based adaptive filter. *IEEE Transactions on Image Processing*, 15(9):2820–2830, 2006.

[84] Tomoo Mitsunaga and Shree K Nayar. Radiometric self calibration. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1. IEEE, 1999.

[85] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[86] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a" completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.

[87] Hossein Ziaei Nafchi, Atena Shahkolaei, Reza Farrahi Moghaddam, and Mohamed Cheriet. Fsitm: A feature similarity index for tone-mapped images. *IEEE Signal Processing Letters*, 22(8):1026–1029, 2015.

[88] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[89] Mansour Nejati, Maryam Karimi, SM Reza Soroushmehr, Nader Karimi, Shadrokh Samavi, and Kayvan Najarian. Fast exposure fusion using exposedness function. In *IEEE International Conference on Image Processing*, pages 2234–2238, 2017.

[90] Tae-Hyun Oh, Joon-Young Lee, Yu-Wing Tai, and In So Kweon. Robust high dynamic range imaging by rank minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1219–1232, Jun. 2015.

[91] Fabrizio Pece and Jan Kautz. Bitmap movement detection: HDR for dynamic scenes. In *IEEE Conference on Visual Media Production*, pages 1–8, 2010.

[92] Photomatix, 2017. Commercially-Available HDR Processing Software.

[93] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 6, 2017.

[94] K Ram Prabhakar, V Sai Srikar, and R Venkatesh Babu. Deepfuse: A deep unsupervised approach for exposure fusion with extreme exposure image pairs. In *IEEE International Conference on Computer Vision*, pages 4724–4732. IEEE, 2017.

[95] Xiameng Qin, Jianbing Shen, Xiaoyang Mao, Xuelong Li, and Yunde Jia. Robust match fusion using optimization. *IEEE Transactions on Cybernetics*, 45(8):1549–1560, Aug. 2015.

[96] Shanmuganathan Raman and Subhasis Chaudhuri. Bilateral filter based compositing for variable exposure photography. In *Proc. Eurographics*, 2009.

[97] Erik Reinhard and Kate Devlin. Dynamic range reduction inspired by photoreceptor physiology. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):13–24, 2005.

[98] Erik Reinhard, Wolfgang Heidrich, Paul Debevec, Sumanta Pattanaik, Greg Ward, and Karol Myszkowski. *High Dynamic Range Imaging: Acquisition, Display, and Image-based Lighting*. Morgan Kaufmann, 2010.

[99] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Transactions on Graphics*, 21(3):267–276, 2002.

[100] 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.

[101] Pradeep Sen and Cecilia Aguerrebere. Practical high dynamic range imaging of everyday scenes: photographing the world as we see it with our own eyes. *IEEE Signal Processing Magazine*, 33(5):36–44, 2016.

[102] 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 Graphics*, 31(6):203–1, 2012.

[103] Qi Shan, Jiaya Jia, and Michael S Brown. Globally optimized linear windowed tone mapping. *IEEE transactions on Visualization and Computer Graphics*, 16(4):663–675, 2010.

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

[105] Rui Shen, I. Cheng, and A. Basu. QOE-based multi-exposure fusion in hierarchical multivariate Gaussian CRF. *IEEE Transactions on Image Processing*, 22(6):2469–2478, 2013.

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

[107] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[108] Takashi Shibata, Masayuki Tanaka, and Masatoshi Okutomi. Gradient-domain image reconstruction framework with intensity-range and base-structure constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2745–2753, 2016.

[109] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[110] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[111] Mingli Song, Dacheng Tao, Chun Chen, Jiajun Bu, Jiebo Luo, and Chengqi Zhang. Probabilistic exposure fusion. *IEEE Transactions on Image Processing*, 2012.

[112] Yang Song, Gangyi Jiang, Mei Yu, Yun Zhang, Feng Shao, and Zongju Peng. Naturalness index for a tone-mapped high dynamic range image. *Applied Optics*, 55(35):10084–10091, 2016.

[113] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 5, 2017.

[114] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *IEEE International Conference on Computer Vision*, pages 4809–4817. IEEE, 2017.

[115] Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic images. *IEEE Computer graphics and Applications*, 13(6):42–48, 1993.

[116] Jack Tumblin and Greg Turk. Lcis: A boundary hierarchy for detail-preserving contrast reduction. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 83–90. ACM Press/Addison-Wesley Publishing Co., 1999.

[117] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004.

[118] Gregory J Ward. The radiance lighting simulation and rendering system. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 459–472. ACM, 1994.

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

[120] Shiqian Wu, Lingxian Yang, Wangming Xu, Jinghong Zheng, Zhengguo Li, and Zhijun Fang. A mutual local-ternary-pattern based method for aligning differently exposed images. *Computer Vision and Image Understanding*, 152:67–78, 2016.

[121] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–252, 2015.

[122] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l 0 gradient minimization. In *ACM Transactions on Graphics*, volume 30, page 174. ACM, 2011.

[123] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on Image Processing*, 19(11):2861–2873, 2010.

[124] Famao Ye, Yanfei Su, Hui Xiao, Xuqing Zhao, and Weidong Min. Remote sensing image registration using convolutional neural network features. *IEEE Geoscience and Remote Sensing Letters*, 2018.

[125] Hojatollah Yeganeh and Zhou Wang. Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, 22(2):657–667, 2013.

[126] Kai Zeng, Kede Ma, Rania Hassen, and Zhou Wang. Perceptual evaluation of multi-exposure image fusion algorithms. In *6th International Workshop on Quality of Multimedia Experience*, pages 27–28, 2014.

[127] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on Image Processing*, 2017.

[128] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, 2018.

[129] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 6, 2018.

[130] Lei Zhang, Weisheng Dong, David Zhang, and Guangming Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, 2010.

[131] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic Imaging*, 20(2):023016, 2011.

[132] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011.

[133] Wei Zhang and Wai-Kuen Cham. Gradient-directed multiexposure composition. *IEEE Transactions on Image Processing*, 21(4):2318–2323, 2012.

[134] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. *arXiv preprint arXiv:1807.02758*, 2018.