2014

Image enhancement methods and applications in computational photography

Qinchun Qian
Louisiana State University and Agricultural and Mechanical College, qinchunqian@gmail.com

IMAGE ENHANCEMENT METHODS AND APPLICATIONS IN COMPUTATIONAL PHOTOGRAPHY

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Electrical and Computer Engineering

by
Qinchun Qian
B.S., Shandong University, 2006
M.E., Nankai University, 2009
M.S., Louisiana State University, 2011
May 2014

To my parents: Zhonglan Xu and Xianming Qian
To my wife: Fuqiong Zhao

Acknowledgements

This dissertation would not be possible without several contributions. I would like to give my first thank to my academic advisor Dr. Bahadir K. Gunturk. The dissertation would not be possible without his invaluable inspiration, encouragement and financial support. My deep gratitude also goes to Dr. Jianhua Chen, Dr. Catherine Deibel, Dr. Xin Li and Dr. Suresh Rai for your time and support to be my committee members. Dr. Jianhua Chen's course Machine Learning and Dr Xin Li's course 3D Graphics and Modeling provided me invaluable knowledge in my research.

Special thank also goes to the Division of Electrical and Computer Engineering for financially supporting my PhD study and providing me good environment for my study. Last but not the least, I would like to thank my family: my mother Zhonglan Xu, my father Xianming Qian, my parents-in-law Aiyun Li and Hanyun Zhao, my older brother Qinhai Qian, my sister-in-law Jie Gu, my nephew Rui Qian for your encouragement and love during the tough times. Thank also goes to my loving wife Fuqiong Zhao, no words in the world can express my gratitude to you for your love and sacrifice during these years.

Table of Contents

Abstract

Computational photography is currently a rapidly developing and cutting-edge topic in applied optics, image sensors and image processing fields to go beyond the limitations of traditional photography. The innovations of computational photography allow the photographer not only merely to take an image, but also, more importantly, to perform computations on the captured image data. Good examples of these innovations include high dynamic range imaging, focus stacking, super-resolution, motion deblurring and so on. Although extensive work has been done to explore image enhancement techniques in each subfield of computational photography, attention has seldom been given to study of the image enhancement technique of simultaneously extending depth of field and dynamic range of a scene. In my dissertation, I present an algorithm which combines focus stacking and high dynamic range (HDR) imaging in order to produce an image with both extended depth of field (DOF) and dynamic range than any of the input images. In this dissertation, I also investigate super-resolution image restoration from multiple images, which are possibly degraded by large motion blur. The proposed algorithm combines the super-resolution problem and blind image deblurring problem in a unified framework. The blur kernel for each input image is separately estimated. I also do not make any restrictions on the motion fields among images; that is, I estimate dense motion field without simplifications such as parametric motion. While the proposed super-resolution method uses multiple images to enhance spatial resolution from multiple regular images, single image super-resolution is related to techniques of denoising or removing blur from one single captured image. In my dissertation, space-

varying point spread function (PSF) estimation and image deblurring for single image is also investigated. Regarding the PSF estimation, I do not make any restrictions on the type of blur or how the blur varies spatially. Once the space-varying PSF is estimated, space-varying image deblurring is performed, which produces good results even for regions where it is not clear what the correct PSF is at first. I also bring image enhancement applications to both personal computer (PC) and Android platform as computational photography applications.

1 Introduction

1.1 Overview of computational photography

The field of computational photography seeks to escape what's impossible in a traditional camera. The innovation of computational photography over the last decade has been remarkable. Compared to traditional photography which merely allows the photographer to take perceived image data, computational photography brings revolutionary change to photography by performing computations on the image data. This change goes beyond the limitations of a traditional camera. As Hayes [1] pointed out, new cameras don't just capture photos, they compute images as well. Nayar [2] explained the computational camera as exploring several dimensions of imaging, including spatial resolution, temporal resolution, field of view, dynamic range, depth and so on. Extensive research has been done by academic and industrial research teams around the world to explore one or more of these dimensions of imaging to enhance image quality. More readings or literature about computational photography may be referred to [3, 4, 5].

We have seen a large number of computational photography applications in market as well. The emerging market of computational photography shows that intensive exploration of image enhancement techniques in computational photography has been motivated by research teams in industry from the marketing point of view. This is achieved by designing new applied optics or image sensors on camera at the hardware level or developing computer vision, computer graphics or image processing techniques at the software level. A good survey of techniques of computational photography is given by Szeliski [5]. Some of good examples

include wavefront coding [6] for extending depth of field, coded aperture [7] for recovering latent image or depth, light field imaging [8] for digital refocusing, recovering radiance maps [9, 10, 11] for high dynamic range imaging, fluttered shutter [12] for removing motion blur, combining multiple images for super-resolution [13, 14, 15, 16, 17, 18] or focus stacking [19, 20, 21, 22, 23, 24]. The output of these techniques is always a regular photo. However, different from photos from a traditional camera, one or more dimensions of imaging are enhanced, which is impossible in a traditional camera. For example, the real world can produce a much larger dynamic range than the two orders of magnitude that a traditional camera can capture. No matter how we tune exposure settings in the camera, including aperture and shutter speed, only partial dynamic range of the scene can be captured in a single image. High dynamic range (HDR) imaging seeks to capture the full dynamic range of a scene by merging multiple images with different exposures. Some techniques such as super-resolution are highly desirable in the fields of medical imaging, biological imaging, high-definition (HD) TV and surveillance system as well.

1.2 Motivation and outline of my dissertation

Although great efforts have been made to enhance one or more dimensions of imaging in the computational photography field, there still remain many challenging issues to be solved in each subfield of computational photography. I will discuss these challenging issues in greater depth in chapters that follow. Particularly, with the popularity of the mobile phone, which is the forefront of computational photography, it brings new challenges to computational photography, as the mobile phone has its specific characteristics such as limited power and memory for image processing compared with its counterparts such as a personal computer or even a traditional digital camera. These issues motivate me to investigate novel computational photography algorithms and to bring new functionalities and experiences to mobile phones. In my dissertation, I will provide some basic knowledge of camera optics and image

sensor for understanding the proposed algorithms; however, I will not extensively cover computational optics, sensors or cameras. Instead, I will put emphasis on software level through employing computer vision and image processing techniques to enhance image quality in computational photography.

The dissertation consists of several projects [25, 26, 27] I explored during my PhD study. My dissertation is outlined as follows:

The aim of imaging is to create accurate, high quality imagery which truly represents the scene. A natural question about image quality is: how to assess the image quality? Actually, image quality factors include sharpness which determines the amount of details the perceived image can convey, dynamic range which is the range of radiance that a camera can capture and noise which is visible random grain in the perceived image, among many other factors. This also holds in the computational photography field. Nayar [2] viewed imaging as having several dimensions, including dynamic range, spatial resolution, depth of field and so on. The computational cameras presented in extensive literature can be explained as explorations of these dimensions of imaging. In Chapter 2, I will present background knowledge and techniques of some of these dimensions of imaging in computational photography which are also essentially important for understanding our proposed image enhancement methods in computational photography. I first present single image deblurring which is aimed at removing blur from single image by estimating blur kernel and restoring the image. Then I go to the super-resolution technique the goal of which is to enhance spatial resolution from multiple images. In the following parts, focus stacking and HDR imaging techniques will be investigated to extend depth of field and dynamic range of a scene, respectively. Also, in each section of this chapter, I will provide some implementations of image enhancement techniques on personal computer or Android phone as part of image enhancement applications in computational photography.

Super-resolution is a technique that enables us to have enhanced spatial resolution and less noise of the scene by merging multiple images, each captured with sub-pixel position shift from other images. We may find extensive literature [13, 14, 15, 16, 17, 18, 28] in super-resolution; however, there are still open problems and challenges that need to be researched further. One of the most important challenges in multi-frame super-resolution restoration has been the motion estimation problem. A second challenge in super-resolution restoration is blur kernel estimation. The majority of existing super-resolution algorithms assume identical blur kernel for all input images; the blur kernel is typically modeled as a symmetric Gaussian function, whose standard deviation is estimated empirically or by some parametric estimation method. The assumption of identical blur kernel does not hold when there are fast moving objects in the scene or the camera is shaken during the exposure time. Ideally, we should estimate the blur kernel for each input image separately. There are few super-resolution methods that estimate blur kernel for each image. Chapter 3 will investigate super-resolution image restoration from multiple images, which are possibly degraded by large motion blur. The blur kernel for each input image is separately estimated. This is unlike many existing super-resolution algorithms, which assume identical blur kernel for all input images. I also do not make any restrictions on the motion fields among images; that is, I estimate dense motion fields without simplifications such as parametric motion. The proposed framework includes three steps: (1) single image motion deblurring using TV-L1 model, (2) optical flow estimation between deblurred images, and (3) weighting-based super-resolution (SR) restoration. The proposed framework is based on our modeling the image formation process as a two-step process. In the first step, low-resolution images are generated from a high-resolution image. In the second step, a single-image blur process describes how to get blurred image from the low resolution image obtained in the first step.

During last decade, we have seen highly successful blind image deblurring algorithms that can even handle large motion blurs; the main underlying factor in the success of these

algorithms is sparse image and/or kernel modeling. Most of these algorithms assume that the entire image is blurred by a single blur kernel just as in our Chapter 3. This assumption does not hold if the scene is not planar or when there are multiple objects moving differently in the scene. In Chapter 4 of this dissertation, I present a method for space-varying point spread function (PSF) estimation and image deblurring from one single image. Regarding the PSF estimation, I do not make any restrictions on the type of blur or how the blur varies spatially. That is, the blur might be, for instance, a large (non-parametric) motion blur in one part of an image and a small defocus blur in another part without any smooth transition. Once the space-varying PSF is estimated, space-varying image deblurring is performed, which produces good results even for regions where it is not clear what the correct PSF is at first. The method has three main steps: (1) Coarse PSF estimation, (2) PSF refinement through kernel clustering, and (3) Space-varying deblurring through image fusion.

Chapter 5 investigates the problem of how to simultaneously extend dynamic range and depth of field of a scene from multiple images with different exposure and focus settings. High dynamic range imaging is a technique of merging multiple images, each captured with different exposure settings to represent partial dynamic range of the scene, to boost the dynamic range of the scene in the created HDR image, whereas focus stacking is the one that merges multiple images, each captured with different focus settings to record partial sharp information of the scene, to extend the depth of field (DOF) in the final created image. However, attention has seldom been given to the problem of simultaneously enhancing two dimensions of imaging: dynamic range and depth of field in the output image. In this chapter, I will present an algorithm which combines focus stacking and HDR imaging in order to produce an image with both extended DOF and dynamic range than any of the input images. The proposed algorithm includes three main steps: (1) spatial registration, (2) focus stacking under exposure diversity, and (3) HDR radiance estimation and tone mapping. In the spatial registration step, images are first photometrically registered using an algorithm

that is insensitive to small geometric variations, and then geometrically registered using an optical flow algorithm. The step of focus stacking under exposure diversity obtains all-in-focus images from registered images with different in-focus regions and exposure information, and lastly, created all-in-focus images in the step of focus stacking under exposure diversity are merged to estimate the HDR radiance map to extend dynamic range. The final created image has both extended DOF and dynamic range.

In Chapter 6, I will conclude our dissertation work and discuss future work of image enhancement methods in computational photography as well.

Our major contributions of this dissertation include:

1. Is it possible to enhance two dimensions of imaging simultaneously in computational photography? This is made possible in my exploration of extending both dynamic range and depth of field of a scene by merging multiple images with different exposure and focus settings.

2. Is it possible to enhance spatial resolution of a scene by merging multiple images which are degraded by large motion blur and which have no restrictions on motion fields? This is also made possible by proposing a unified framework of super-resolution problem and single image motion deblurring problem and by employing state-of-the-art computer vision and image processing techniques for optical flow estimation and blind image deblurring.

3. Is it possible to remove blur from a single image degraded by space-varying blur across the entire image? This is also made possible by combining computer vision and machine learning techniques to release some restrictions on this space-varying image deblurring problem.

4. Is it possible to bring image enhancement applications in computational photography to mobile phone? I illustrate some computational photography applications, including

focus stacking, HDR creation and super-resolution on Android phone. The success shows it is possible to extend new computational photography applications to mobile phones.

2 Background

2.1 Single image deblurring

In digital photography, one of the most common degradations has been image blur. Image deblurring which has been attracting researchers' attention for decades still remains a challenging and open problem. The cause of blur includes moving objects, defocus and camera shake when taking a photo of a scene during the exposure time. Next I will do a literature review of these three main causes of blur in details.

Defocus blur is caused by either the camera's limited depth of field (DOF) or lens aberrations [29]. For both limited DOF and lens aberrations, the observed image can be modeled as convolution of in-focus image of the scene and a Gaussian blur kernel denoting the effect of defocus blur. Many techniques have been proposed to estimate the blur due to defocus[29, 30, 31, 7, 32]. Bae [30] estimated the space-varying defocus blur across the image to obtain the defocus blur map, and then magnified the existing blurriness while keeping sharp regions sharp for photographs such as portraits. On the contrary, instead of magnifying defocus blur for some artistic effects, a majority of deblurring algorithms focus on removing defocus blur. Tai [29] presented a simple approach to estimate defocus blur map to segment in-focus regions from depth of field image and to rank the image quality by introducing a novel local contrast prior. Shen [32] used a modified local contrast prior to derive a defocus blur map and then proposed an image deblurring method to remove the blur and lastly, according to the estimated blur map, the all-in-focus image is constructed by selecting pixels from a set of candidate deblurred images. Tai and Tang [31] proposed

a technique for recovering image details that are lost due to defocus blur. The proposed algorithm took advantage of image data and prior to obtain more visually desired results. Different from the above software techniques for removing defocus blur, Levin [7] proposed a joint hardware and software solution that allowed for recovering both all-in-focus image and extracting depth from a single image taken by a conventional camera with a coded aperture.

The motion blur caused by camera shake is very common in digital photography because it is difficult to hold the camera to be sufficiently steady. Commonly, camera shake is modeled as a blur kernel, which describes the camera movement during the exposure time. Significant progress has been made towards blindly removing uniform blur kernel from a single image or multiple images. Among them, great success has been made in estimating a complex motion blur from a single image [33, 34, 35, 36, 37, 38, 39, 40, 41]. Sparse priors [33, 34, 35] instead of Gaussian priors are employed for either blur kernel or latent image, leading to the sparse representation for the blur kernel and sharp representation for the deblurred latent image. On the other hand, sharp edge prediction step [36, 37] is introduced for the motion blur estimation instead of using sparse priors. Xu [38] proposed a two-phase kernel estimation in which a new metric is introduced to measure the usefulness of image edges for kernel estimation based on spatial prior and kernel refinement. However, the above success of single-image motion deblurring is limited due to insufficient information from one single image for estimating some large and complex blur PSFs. Chen [39], Li [40] and Yuan [41] combined information extracted from either multiple successive images or stereo images to robustly estimate blur PSF.

However, these motion deblurring algorithms for uniform blur kernel caused by camera shake have its limitation due to the fact that the real camera shake does not cause uniform blur kernel, leading to challenging space-varying motion deblurring problems due to camera shake. Harmeling [42] introduced a taxonomy of camera shake and proposed an algorithm for simultaneously estimating motion blur and obtaining deblurred images based on the

built experiment setup. However, the main limitations of this approach are that it can only deal with small blur kernels and that the space-varying motion blurs should vary smoothly across the image. Joshi [43] proposed an algorithm to estimate space-varying PSFs due to defocus, slight camera shake or inherent aspects of the camera imaging system. The algorithm predicted a sharp version of the blurred image and used both sharp and blurred versions to estimate the blur PSF. However, the method may fail if the blur kernel to be estimated has multiple peaks. Whyte [44] introduced a parameterized geometric model of the blurring process which explained the non-uniform motion blur caused by camera shake in terms of the rotational velocity of the camera during exposure. However, the proposed model is only applicable for static scenes. Xu and Zheng [45] proposed a new L0 sparse expression which demonstrated fast convergence for motion deblurring. In addition, the proposed framework is suitable for both uniform and non-uniform motion blur removal. Rather than using a single image in the above discussed non-uniform blur removal algorithms, Xu and Jia [46] employed a stereopsis configuration where two stereo images were captured to tackle the space-varying motion deblurring problem in a scene with depth variation. In cases where the depth change is not critical to the variation of the blur kernel, the proposed method can also be applied to single-image motion deblurring problems.

Object motion during exposure is another main cause of image blur. Part of the image may be blurred by the moving objects in the scene. In such situations, a single uniform blur cannot model the motion blur caused by the moving object across the whole image. Levin [47] assumed that the moving objects moved with a constant velocity and the image was blurred by one-dimensional rectangular blur. The proposed algorithm first estimated the direction of motion as the direction with minimal variation of derivatives. Then the image was segmented based on the statistics of derivatives expected in an average natural image. Liu [48] locally detected and classified motion blur using a set of cues including gradient histogram, maximum saturation and variance of autocorrelation function in different directions. The

image was classified as blurred and non-blurred regions based on the decision of a Bayes classifier. Chakrabarti [49] presented a novel local blur cue to measure the likelihood of a small neighborhood blurred by a candidate blur kernel, and proposed an algorithm which simultaneously selected a motion blur kernel and segmented the regions that it affected. However, the motion blur kernel was assumed to be one of horizontal or vertical box filters with certain lengths. Shan [50] proposed rotational motion deblurring algorithm which could be applied to situations where a single image was degraded by a blurred object due to the rotational and translational motions. The proposed algorithm introduced a transparency map encoding the object motion and estimating the motion parameters. Cho [51] utilized multiple images that were degraded by space-varying motion blurs to simultaneously restore images and to estimate associated motion blur kernels, image segmentation and motion as well. The common drawback of these proposed space-varying motion deblurring algorithms assumes that the motion blur caused by moving object has some parameterized pattern. For example, the motion blur may be only one-dimensional rectangular filter or has either horizontal or vertical direction.

In the following parts, I will first address the image deblurring problem with a linear and shift-invariant (LSI) degradation model and then present some commonly used image deblurring methods.

2.1.1 Image blur model of linear and shift-invariant (LSI) system

Although the image degradation process is generally nonlinear and space varying, the LSI model provided a good starting point for understanding the image delburring problem. In addition, LSI model is also regarded as a special case of space-varying image deblurring problems. The blur process is modeled as follows

$$g = f \otimes h + n, \tag{2.1}$$

where g is the blurred image, f is the latent image, h is the blur PSF, $\otimes$ operator denotes convolution and n is the additional noise. Figure 2.1 shows single image blur process for

linear and shift-invariant model. Noting that convolution is a linear operator, and thus blur process can also be written as

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \tag{2.2}$$

where $\mathbf{H}$ is a convolution matrix, $\mathbf{g}$, $\mathbf{f}$ and $\mathbf{n}$ are vectors denoting the blurred image g, the latent image f and the additional noise n, respectively. In the following part of my dissertation, I may use matrix or vector notations alternatively.

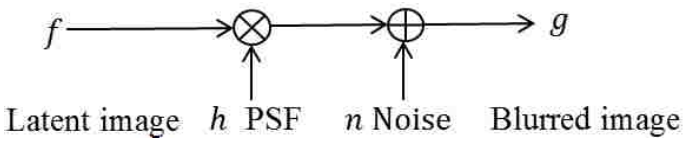


FIGURE 2.1. Single image blur process for linear and shift-invariant model.

Image deblurring in LSI is called image deconvolution. If the blur kernel PSF is known, the image deblurring is called non-blind image deconvolution, otherwise, image deblurring is called blind image deconvolution. In the blind deconvolution, the blur kernel PSF needs to be estimated first and then estimate latent image or the blur kernel PSF and latent image are jointly estimated. Here, an improved version of Cho's method [37] is utilized to estimate blur PSF and then some commonly used non-blind image restoration methods are investigated.

2.1.2 Point spread function (PSF) estimation

The coarse PSF estimation adopts coarse-to-fine image pyramid strategy to progressively refine the blur kernel. In each pyramid, the estimation method iterates three steps: (I) strong edge prediction, (II) fast kernel estimation, and (III) coarse image deconvolution.

I. Strong edge prediction

In the strong edge prediction step, image gradient map $\nabla \tilde{f}^s$ of the latent image f is obtained by keeping only strong edge gradients. This step consists of Gaussian filtering, shock filtering and gradient magnitude thresholding. Gaussian filtering is first applied to the latent image f to eliminate possible noise and small details. A shock filter [52] is then applied to the Gaussian filtered latent image to restore strong edges. However, the shock

filtered image $\tilde{f}$ also contains enhanced noise. A threshold is calculated and applied to the gradient maps $\nabla\tilde{f}$ of $\tilde{f}$ to eliminate the noise. Here, the threshold for truncating the gradient is set as fixed value by trial and error for simplicity, resulting in strong image gradient maps $\nabla\tilde{f}^s$.

II. Fast kernel estimation

In this step, blurred image g and predicted strong edge $\nabla\tilde{f}^s$ obtained from the first step are used for kernel estimation. Instead of using sparse prior for regularization term, Gaussian prior is included in the following energy function:

$$E(h) = \|\nabla g - \nabla\tilde{f}^s \otimes h\|^2 + \gamma\|h\|^2, \tag{2.3}$$

where γ is a weight to balance the data fidelity term and regularization term. Here, it is set as 10. A closed-form solution is obtained through FFT and inverse FFT operations when optimizing the energy function, resulting in fast computational speed. The closed-form solution for h is given by

$$h = F^{-1}\left(\frac{\overline{F(\tilde{f}_x^s)}F(g_x) + \overline{F(\tilde{f}_y^s)}F(g_y)}{\overline{F(\tilde{f}_x^s)}F(\tilde{f}_x^s) + \overline{F(\tilde{f}_y^s)}F(\tilde{f}_y^s) + \gamma}\right), \tag{2.4}$$

where $F()$, $F^{-1}()$ and $\overline{F()}$ denote FFT, inverse FFT and complex conjugate FFT operations.

III. Coarse image deconvolution

In this deconvolution step, the latent image f is estimated from the above estimated kernel h and the input blurred image g, instead of using image gradient $\|\nabla f\|^2$ as Gaussian prior to guide the restoration of the latent image, a new spatial prior $\|\nabla f - \nabla\tilde{f}^s\|^2$ (see Xu [38]) is used which prefers a sharp recovery even though we use the Gaussian regularization term. The resulting energy function is

$$E(f) = \|g - f \otimes h\|^2 + \lambda\|\nabla f - \nabla\tilde{f}^s\|^2, \tag{2.5}$$

where weight λ is set to be 0.1. The equation can be optimized very fast again through FFT and inverse FFT operations, the coarse version of the latent image f is obtained as

$$f = F^{-1}\left(\frac{\overline{F(h)}F(g) + \lambda(\overline{F(\nabla x)}F(\tilde{f}_x^s) + \overline{F(\nabla y)}F(\tilde{f}_y^s))}{\overline{F(h)}F(h) + \lambda(\overline{F(\nabla x)}F(\nabla x) + \overline{F(\nabla y)}F(\nabla y))}\right), \tag{2.6}$$

where ∇x and ∇y denote image gradient operators in the horizontal and vertical directions, respectively. Finally, blur kernel sparsity is enforced through hard-thresholding, which sets the kernel values that are less than the 10 percent of the maximum value in the kernel to be 0. An example will show the iteration process of how to progressively refine the blur kernel from top to bottom. The blurred image is with the courtesy of Jia [38].

2.1.3 Non-blind image restoration methods

In the last section, blur PSF h is estimated, in this section, the latent image $\mathbf{f}$ will be restored based on estimated blur PSF. In the image restoration problem, the latent image $\mathbf{f}$ and additional noise $\mathbf{n}$ are viewed as random variables in the image blur process, Bayesian estimation can be used to convert image restoration problem into maximum a posterior (MAP) estimator [7]. The MAP estimator may be written as:

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathbf{g}) = \arg\max_{\mathbf{f}} \frac{p(\mathbf{g}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{g})} = \arg\max_{\mathbf{f}} p(\mathbf{g}|\mathbf{f})p(\mathbf{f}) \tag{2.7}$$

Since $p(\mathbf{g})$ is constant with respect to latent image $\mathbf{f}$, $p(\mathbf{g})$ was dropped in the above equation. In the equation for MAP estimator, $p(\mathbf{g}|\mathbf{f})$ is denoted as noise prior and $p(\mathbf{f})$ is denoted as image prior. Next different combinations of image prior and noise prior will be investigated [53, 54, 7, 55], leading to different image deblurring models.

I. No image prior, Poisson noise prior

Richardson-Lucy deconvolution [53, 54] is a well-known image restoration algorithm to recover the latent image blurred by a known PSF. The statistics are performed under the assumption that noise is Poisson distributed, which is appropriate for photon noise in the data, this leads to an equation for f which can be iteratively solved according to the following update function which starts with arbitrary initial estimate f^0:

$$f^{t+1} = f^t(\frac{g}{f^t \otimes h} \otimes h^{flip}), \tag{2.8}$$

where the division and multiplication operations are element wise, and h^{flip} is the flipped version of PSF h. That is, $h^{flip}_{mn} = h_{(i-m)(j-n)}, 0 \leq m \leq i, 0 \leq n \leq j$. It has been shown that

if the iteration process converges, it will converges to the Maximum-likelihood (ML) solution for f [56]. MATLAB build-in function *deconvlucy* can be used to restore the latent image given blurred image and known PSF using Richardson-Lucy algorithm.

II. No image prior, Gaussian noise prior

In this case, The MAP estimator is reduced to maximum likelihood (ML) estimator. If the noise is Gaussian distributed, the ML estimator may be rearranged as follows:

$$\hat{f} = \arg\max_f \frac{1}{\sqrt{(2\pi)^N}} \exp(\frac{-1}{2}\|g - f \otimes h\|^2) \tag{2.9}$$

By taking logarithm on the above equation, the ML estimator is reduced to least square (LS) problem:

$$\hat{f} = \arg\min_f \|g - f \otimes h\|^2 \tag{2.10}$$

By looking at the above LS problem, convolution operation in the spatial domain can be performed as multiplication operation in the frequency domain. Specifically, FFT and inverse FFT are used to perform image restoration efficiently as follows:

$$f = F^{-1}\left(\frac{\overline{F(h)}F(g)}{\overline{F(h)}F(h)}\right), \tag{2.11}$$

Without image prior, the deconvolution process is sensitive to noise and leads to unstable solution. This is easily seen when $F(h)$ is very small from the above equation.

III. Gaussian image prior, Gaussian noise prior

By introducing image prior, the solution will be pulled towards zero when $F(h)$ is very small. Instead of using image intensity distribution, image gradient is widely chosen as image prior. With Gaussian image prior, the image gradient distribution of latent image f is as follows:

$$p(f) = \frac{1}{\sqrt{(2\pi)^N}} \exp(\frac{-1}{2}\|\nabla f\|^2), \tag{2.12}$$

where ∇ denotes image gradient operator. Gaussian image prior implies that the solution prefers to choose image gradient to be smoothly distributed across the entire image. In this

case, the MAP estimator is reduced to the following minimization problem:

$$\hat{f} = \arg\min_{f} \|g - f \otimes h\|^2 + \lambda \|\nabla f\|^2 \tag{2.13}$$

The above minimization problem is least square problem with Tikhonov regularization where λ is the regularization parameter to balance the two terms in this minimization problem; either spatial domain or frequency domain solution can be given to minimize the objective function. In spatial domain, gradient decent or conjugate gradient method can be used to efficiently approach to the global optimum, however, this is not guaranteed to achieve the optimum. In frequency domain, FFT and inverse FFT can be used to get the closed-form solution as follows:

$$f = F^{-1}\left(\frac{\overline{F(h)}F(g)}{\overline{F(h)}F(h) + \lambda(\overline{F(\nabla x)}F(\nabla x) + \overline{F(\nabla y)}F(\nabla y))}\right), \tag{2.14}$$

IV. Sparsity image prior, Gaussian noise prior

The advantage of Gaussian image prior is that optimal closed-form solution can be achieved by using FFT and inverse FFT operations. On the other hand, Gaussian image prior leads to smooth image; however, image details are around edge regions. Then, sparsity image priors are introduced by Levin [7] which opt to concentrate image gradient on a small number of image pixels, leaving the majority of image gradient to be small. The sparsity image prior leads to sharper edges. The model of sparsity image prior is written as follows:

$$p(f) = \exp(-\alpha \sum_{x} \sum_{y} \rho(\nabla f_{(x,y)})), \tag{2.15}$$

where a sparse function $\rho(z) = \|z\|^{0.8}$ is used in the model. The resulting objective function is as follows:

$$\hat{f} = \arg\min_{f} \|g - f \otimes h\|^2 + \lambda \sum_{x} \sum_{y} \rho(\nabla f_{(x,y)}) \tag{2.16}$$

The disadvantage of a sparse prior is that the objective function is non-convex. Then a closed-form solution cannot be obtained as Gaussian prior. Levin [7] proposed iterative reweighted least square (IRLS) method to iteratively solve this non-convex objective function.

V. Total variation (TV) for image prior, L1 norm for noise model

In some cases, Gaussian model for noise is non-robust to outliers for image deconvolution, L1 norm is used for data fidelity term, while total variation (TV) is used for regularization, leading to TV-L1 model [55]

$$E(f) = \|g - f \otimes h\|_1 + \lambda \|\nabla f\|_2, \tag{2.17}$$

Clearly, it brings non-linearity and non-differentiability to both data and regularization terms, resulting in computational difficulty to solve this problem. To overcome the computational difficulties, Wang [55] proposes an alternating minimization method based on half quadratic splitting to solve the TV-L1 optimization problem. In [55], auxiliary variables are introduced to solve this optimization problem. The auxiliary variables ν for $g - f \otimes h$ and ω for ∇f are introduced, leading to a modified objective function as follows:

$$\begin{aligned} E(f, \omega, \nu) &= (\frac{1}{2\beta}\|g - f \otimes h + \nu\|_2^2 + \|\nu\|_1) \\ &+ \lambda\|\omega\|_2 + \frac{1}{2\theta}\|\nabla f - \omega\|_2^2. \end{aligned} \tag{2.18}$$

When $\beta \to 0$ and $\theta \to 0$, the solution of (2.17) will approach to that of (2.16). Here an alternating minimization method is adopted to iteratively solve the objective function (2.17) with respect to f, ω, ν. At each iteration, given initial or estimated ω, ν, the following objective function is first solved by fixing ω, ν:

$$E(f; \omega, \nu) = (\frac{1}{2\beta}\|g - f \otimes h + \nu\|_2^2) + \frac{1}{2\theta}\|\nabla f - \omega\|_2^2. \tag{2.19}$$

The above quadratic function is actually a least squares problem with generalized Tikhonov regularization. In [55], a closed-form solution is obtained based on Fast Fourier Transforms (FFTs)

$$f = F^{-1}\left(\frac{\overline{F(h)}F(g+\nu) + \frac{\beta}{\theta}(\overline{F(\nabla x)}F(\omega_x) + \overline{F(\nabla y)}F(\omega_y))}{\overline{F(h)}F(h) + \frac{\beta}{\theta}(\overline{F(\nabla x)}F(\nabla x) + \overline{F(\nabla x)}F(\nabla x))}\right), \tag{2.20}$$

Given the f estimate, because ω and ν belong to different terms, separate and independent objective functions are generated as:

$$E(\omega; f) = \lambda\|\omega\|_2 + \frac{1}{2\theta}\|\nabla f - \omega\|_2^2 \tag{2.21}$$

$$E(\nu; f) = \frac{1}{2\beta}\|g - f \otimes h + \nu\|_2^2 + \|\nu\|_1 \tag{2.22}$$

The optimal solutions for ω and ν are derived according to the shrinkage formula:

$$\omega_x = \frac{\partial_x f}{\|\nabla f\|_2}\max(\|\nabla f\|_2 - \theta\lambda, 0) \tag{2.23}$$

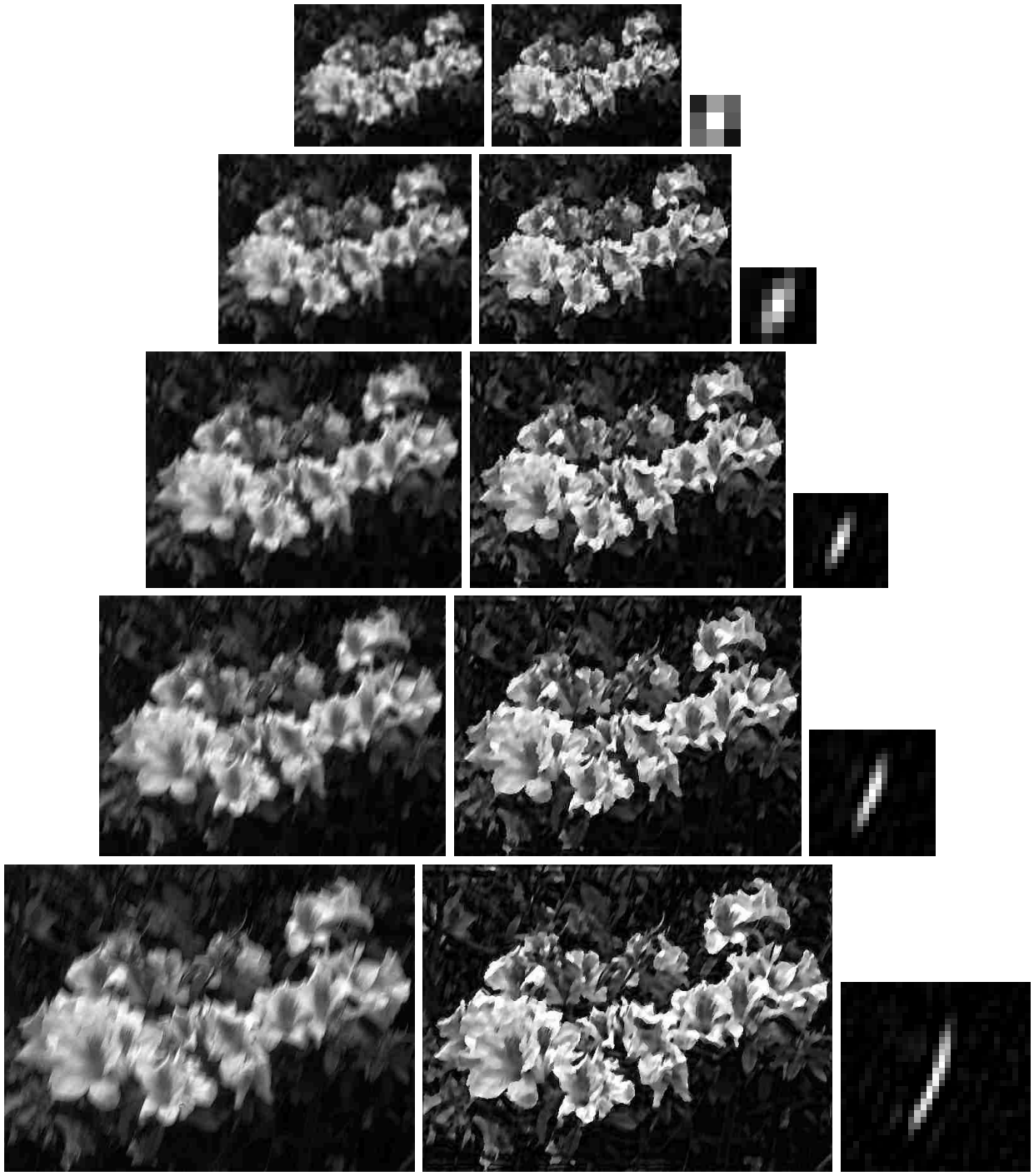
$$\omega_y = \frac{\partial_y f}{\|\nabla f\|_2}\max(\|\nabla f\|_2 - \theta\lambda, 0) \tag{2.24}$$

$$\nu = \text{sign}(g - f \otimes h)\max(\|g - f \otimes h\| - \beta, 0) \tag{2.25}$$

where $\partial_x f$ and $\partial_y f$ are image gradients in the horizontal and vertical directions. In brief, in the alternating minimization algorithm for image restoration problem, f is iteratively computed according to update function (2.19) for fixed ω and ν, and then ω_x, ω_y, ν are computed according to equations (2.22), (2.23), and (2.24) for fixed f. These steps are iterated until convergence.

2.1.4 Implementation of blind image deblurring

I developed a Matlab toolbox to implement blind image deblurring. The blind image deblurring consists of two steps: blur kernel PSF estimation and non-blind image restoration based on estimated blur PSF. In the toolbox, the PSF is progressively refined using coarse-to-fine image pyramid [51]. After estimating PSF, image restoration is performed on blurred image using estimated PSF. Different kinds of models incorporating image prior and noise prior [53, 54, 7, 55] are provided in the toolbox. Figure 2.2 shows the results of blur kernel PSF estimation using Matlab toolbox.



(a) (b) (c)

FIGURE 2.2. Example of blur kernel estimation. Coarse-to-fine pyramid consists of five levels from top to bottom. At each level, from left to right, it includes three images: (a) Blurred image at different pyramid level (b) Coarse deblurred image (c) Estimated blur kernel PSF for that pyramid. (The size of original image is 486×648, at each pyramid, downsampled blurred image is bilinearly downsampled with the downsampling factor 2/3. From top to bottom, the size of blur kernel is 3×3, 7×7, 15×15, 21×21, 31×31.)

Figure 2.3 gives deblurred images using R-L algorithm [53, 54], L2+Tikhonov model, L2+sparsity prior model [7] and TV-L1 model [55], respectively. By comparison, TV-L1 model can efficiently estimate sharp image without artifacts such as ringing in other deblurred images. On the other hand, Levin's L2+sparsity prior model can also recover sharp image without artifacts, however, IRLS method [7] is much more time-consuming that Wang's alternating minimization [55] method in TV-L1 model.

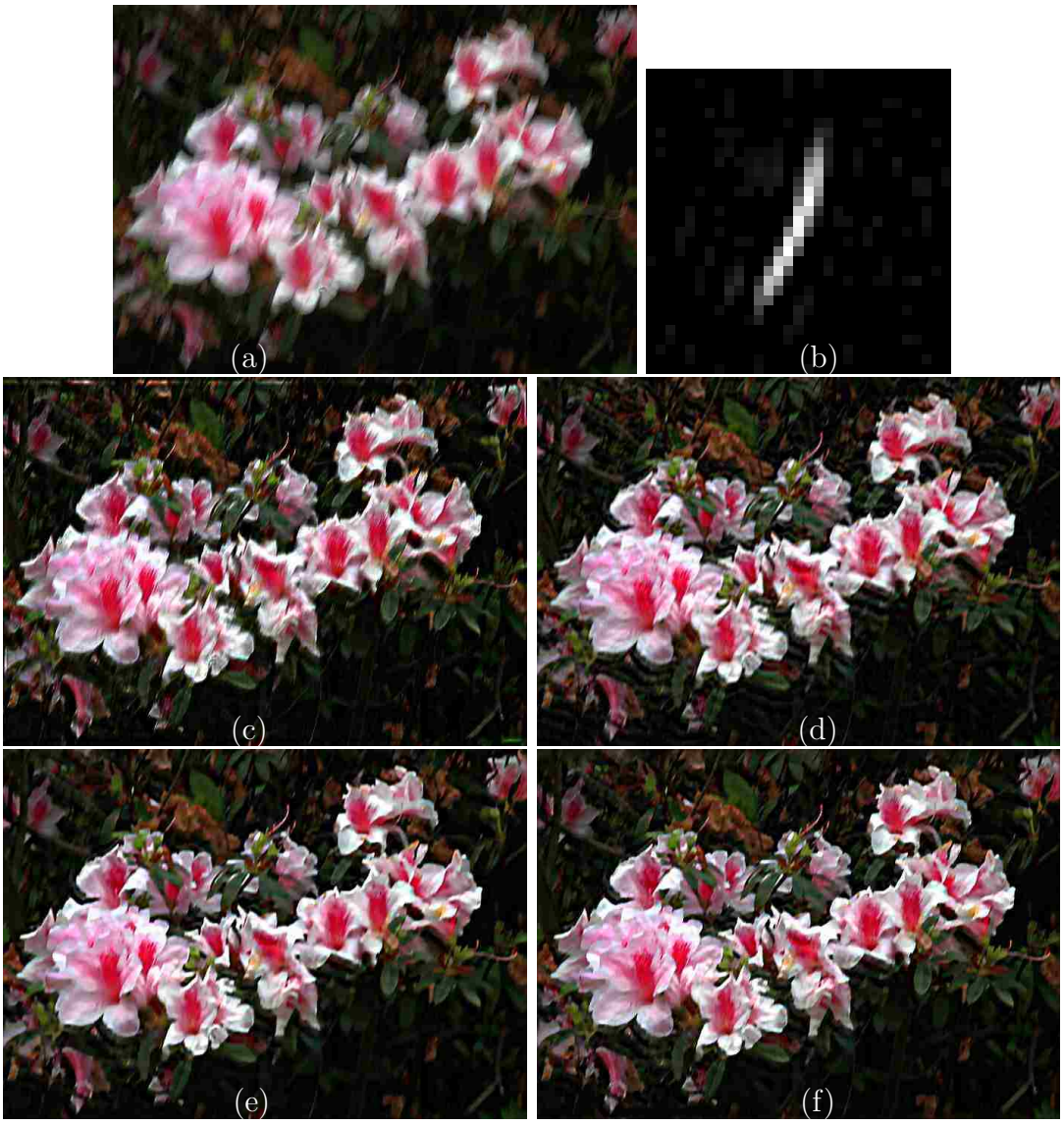


FIGURE 2.3. Example of blind image deblurring. (a) Original blurry image (b) Estimated motion blur kernel (c) Deblurred image using R-L model (d) Deblurred image using L2+Tikhonov model (e) Deblurred image using L2+Sparsity model (f) Deblurred image using TV-L1 model

2.2 Super-resolution from multiple images

Super-resolution (SR) seeks to enhance spatial resolution of a scene from multiple low-resolution (LR) images. The study on SR imaging has been extensively investigated for more than two decades. In addition, super-resolution can find many applications in the fields of medical imaging, surveillance and high-definition (HD) TV. A comprehensive literature review of super-resolution imaging may refer to [13, 14, 15, 16, 17, 18, 28]. The most common way to formulate the super-resolution problem is to first get the image process model of super-resolution which relates the original super-resolution image and multiple low-resolution images, and then to recover the original super-resolution image based on Bayesian estimation. Bayesian estimator considers image process model and original super-resolution image as random variables. Next image process model of super-resolution will be presented and then different SR algorithms based on Bayesian estimation theory will be investigated [57].

2.2.1 Image process model of super-resolution

Most existing SR reconstruction algorithms assume an observation model which relates the high-resolution image and multiple low-resolution images. The observation model [57] includes the main sources of information losses, such as aliasing, blurring and noise. According to the image model of SR, low-resolution images $\{\mathbf{g}_k\}$, $k = 1, ..., N$, are generated from a high-resolution image $\mathbf{f}$ through a sequence of linear operations: image warping $\{\mathbf{M}_k\}$, image blurring $\mathbf{B}_k$ with a point spread function (PSF), and common downsampling $\mathbf{D}$. The imaging process can be formulated as

$$\mathbf{g}_k = \mathbf{D}\mathbf{B}_k\mathbf{M}_k\mathbf{f} + \mathbf{n}_k = \mathbf{H}_k\mathbf{f} + \mathbf{n}_k. \tag{2.26}$$

In most SR algorithms, one of the low-resolution images is used as reference image. The SR image is reconstructed on top of this common grid. Image warping $\{\mathbf{M}_k\}$ is estimated with the help of image registration. Many super-resolution algorithms [14, 15] assume that the motion field is restricted to a parametric model, such as perspective model or even

translational model. This is obviously very limiting and may not be used in most real-life applications. To deal with more realistic motion, block-based motion estimation [58] or optical flow methods [59, 60] could be utilized. In recent years, highly accurate and practical optical flow methods [16, 61] have been proposed; and these optical flow methods definitely improve the performance in super-resolution applications. Schultz [62] used dense motion fields for image warping. Hardie [63, 64] tried to re-estimate the image warping after getting initial sharp image to jointly estimate sharp image and image registration altogether.

In the super-resolution problem, image blurring $\mathbf{B}_k$ may be caused by optical system (defocus blur) or motion (camera shake or moving objects in the scene). Particularly, for motion blur, the PSF may be space-varying across the entire image. The PSF may be assumed to be known from the camera sensor optics or estimated from the low-resolution images. In the section of single image deblurring, I have discussed a lot about PSF estimation techniques. Most of current super-resolution algorithms simply assume a known Gaussian blur kernel. Bascle [65] considered motion blur but using affine motion model. Liu [64] deals with both blur kernel estimation and complex motion under a Bayesian model which estimates high resolution image and PSF jointly, but motion blur is not taken into account into his system.

2.2.2 Super-resolution reconstruction algorithms

The above image process model explained how to get k-th observation image $\mathbf{g}_k$ from original super-resolution image $\mathbf{f}$ and discussed image warping and image blurring issues in the super-resolution problem. All N low-resolution images can be stacked to form a simplified representation of SR problem as follows:

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}, \tag{2.27}$$

where $\mathbf{g} = [\mathbf{g}_0^T \mathbf{g}_1^T \cdots \mathbf{g}_{N-1}^T]^T$, $\mathbf{H} = [\mathbf{H}_0^T \mathbf{H}_1^T \cdots \mathbf{H}_{N-1}^T]^T$ and $\mathbf{n} = [\mathbf{n}_0^T \mathbf{n}_1^T \cdots \mathbf{n}_{N-1}^T]^T$.

Many frequency-based SR algorithms [66, 67, 13] have been proposed to enhance the resolution in the frequency domain. Although frequency-based SR methods are computationally

efficient, they are restricted to translational motion and LSI blur. SR algorithms in spatial domain are more extensively proposed because some of them can incorporate different types of motion fields, PSF and regularization terms. Projection onto convex sets (POCS) [68, 69, 70] is introduced to SR problem that recovers the original SR image which is consistent with predefined data constraint set. However, a majority of SR algorithms are proposed based on the MAP approach. If original super-resolution image $\mathbf{f}$, low-resolution images $\mathbf{g}$ are considered as random variables, the SR problem could be described as MAP estimate according to Bayesian theorem [57]:

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} p(\mathbf{f}|\mathbf{g}) = \arg\max_{\mathbf{f}} \frac{p(\mathbf{g}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{g})} = \arg\max_{\mathbf{f}} p(\mathbf{g}|\mathbf{f})p(\mathbf{f}) \tag{2.28}$$

In the above equation, $p(\mathbf{g}|\mathbf{f})$ describes image process model mismatch, including noise, image warping error or false PSF estimation. Whereas p($\mathbf{f}$) acts as the image prior leading to a stable SR solution. Without an image prior, the MAP estimate is reduced to ML estimate which is equivalent to least square problem without regularization term. The direct solution of least square problem is straightforward but infeasible due to the large size of matrices. Alternatively, iterative approaches such as gradient decent [71], preconditioned conjugate gradient decent [72] are used to solved least squares problem. In order to stabilize the SR solution, image prior is introduced to SR problem. Tikhonov regularization [73] is one of the most common image prior forms. Unfortunately, the Tikhonov regularization term tends to produce oversmooth solutions since this form of image prior assumes that the image gradient is Gaussian distributed. Farsiu [14] provided total variation (TV) for image prior which is claimed to preserve strong edges when solving SR problem. Some other forms of image prior such as robust priors are proposed, which encourage an edge preserving SR solution. These robust priors include Huber function, Lorentzian function and Tukey function [74]. A robust function is used not only in image prior, but also more importantly in image process model. The reason why a robust function is also applied to the data fidelity term is that image process model mismatch always occurs. The factors of image process model mismatch

include noise, false blur PSF estimation or image registration. Thus, robust error functions have been greatly applied to the data fidelity term to handle image process model mismatch in SR reconstruction problem. If image prior is included in the SR problem, the MAP problem we mentioned earlier is reduced to the following cost function to be minimized:

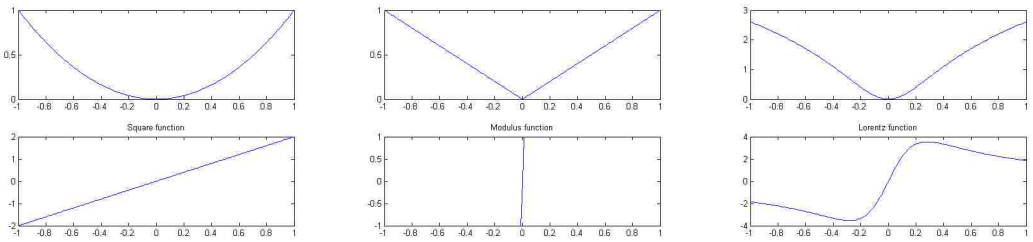
$$C(\mathbf{f}) = \sum_k \mathbf{1}^T \rho(\mathbf{g}_k - \mathbf{H}_k\mathbf{f}) + \lambda \mathbf{1}^T \psi(\mathbf{R}\mathbf{f}), \tag{2.29}$$

where $\mathbf{1}$ is column vector of 1's, $\rho()$ and $\psi()$ are robust functions on data fidelity term and regularization term, respectively. The robust functions are operated element wise, $\mathbf{R}$ is a matrix operation which may be an identity matrix or a matrix for image gradients and so on. As a matter of fact, the above equation is a generic model. Some well-known SR problems are considered as special cases of this generic model. For example, if $\mathbf{R}$ is gradient operator and $\rho(z) = \psi(z) = z^2$, this generic model of SR problem is reduced to well-known least square problem with Tikhonov regularization. If $\rho(z) = \psi(z) = |z|$ and $\mathbf{R}$ is still image gradient operator, this is reduced to another well-known approach: TV-L1 SR problem. However, other robust functions may be selected for both data fidelity term and regularization term. Some functions and their derivatives which are also called influence functions are given in Table 2.1 and plotted in Figure 2.4.

TABLE 2.1. Some functions and influence functions

Name	Function	Influence function
Square function	$\rho(z) = z^2$	$\rho^{'}(z) = 2z$
Modulus function	$\rho(z) = \|z\|$	$\rho^{'}(z) = sign(z)$
Lorentz function	$\rho_\sigma(z) = log(1 + \frac{z^2}{2\sigma^2})$	$\rho_\sigma^{'}(z) = \frac{2z}{2\sigma^2 + z^2}$

In the following part, three typical forms of functions will be investigated, that is, square function, modulus function and robust function, for data fidelity term. For simplicity, square function $\psi(z) = z^2$ will be adopted for regularization term, leading to common Tikhonov term. In order to apply iterative methods such as gradient decent method to minimize the cost function, the functions on both data fidelity term and regularization term should be



(a) (b) (c)

FIGURE 2.4. Some functions and influence functions. (a) Square function and its influence function; (b) Modulus function and its influence function; (c) Lorentz function and its influence function;

differentiable. Starting with an initial estimate $\mathbf{f}^{(0)}$, the updated estimate at the nth iteration is

$$\begin{aligned}\mathbf{f}^{(n+1)} &= \mathbf{f}^{(n)} - \mu\nabla C(\mathbf{f}^{(n)}) \\ &= \mathbf{f}^{(n)} - \mu\sum_k \mathbf{H}_k^T\rho^{'}(\mathbf{g}_k - \mathbf{H}_k\mathbf{f}^{(n)}) - \mu\lambda\mathbf{R}^T\psi^{'}(\mathbf{R}\mathbf{f}^{(n)}),\end{aligned} \tag{2.30}$$

where μ is the step size, which could be empirically set or adaptively adjusted in the experiment. $\rho^{'}()$ and $\psi^{'}()$ may be any of influence functions shown in Table 2.1.

I. L2 norm for data fidelity term, Tikhonov regularization term

If $\mathbf{R}$ is gradient operator and $\rho(z) = \psi(z) = z^2$, the SR problem is reduced to least square problem (L2 norm for data fidelity term) with Tikhonov regularization term. the generic cost function is reduced to

$$C(\mathbf{f}) = \sum_k \|\mathbf{g}_k - \mathbf{H}_k\mathbf{f}\|_2^2 + \lambda\|\nabla\mathbf{f}\|_2^2 \tag{2.31}$$

The updated solution based on gradient decent method is written as:

$$\begin{aligned}\mathbf{f}^{(n+1)} &= \mathbf{f}^{(n)} - \mu\nabla C(\mathbf{f}^{(n)}) \\ &= \mathbf{f}^{(n)} - \mu\sum_k \mathbf{M}_k^T\mathbf{B}_k^T\mathbf{D}^T(\mathbf{g}_k - \mathbf{D}\mathbf{B}_k\mathbf{M}_k\mathbf{f}^{(n)}) - \mu\lambda\nabla^T(\nabla\mathbf{f}^{(n)}),\end{aligned} \tag{2.32}$$

By looking at the influence function $\rho^{'}(z) = 2z$, its influence function assigns larger weights to larger errors and is thus very vulnerable in the presence of outliers.

II. L1 norm for data fidelity term, Tikhonov regularization term

If $\mathbf{R}$ is gradient operator and $\rho(z) = |z|, \psi(z) = z^2$, the SR problem is reduced to L1 norm for data fidelity term with Tikhonov regularization term. As a robust alternative to L2 norm, L1 error norm is much more robust than L2 error norm; however, by looking at influence function, L1 norm for data fidelity does not distinguish between small and large errors because of assigning the same weights to both small and large errors. This may suppress large amount of outliers, on the other hand, it eliminates useful high-frequency information as well. The corresponding cost function for L1 data fidelity term with Tikhonov Regularization term is written as:

$$C(\mathbf{f}) = \sum_k \|\mathbf{g}_k - \mathbf{H}_k\mathbf{f}\|_1 + \lambda\|\nabla\mathbf{f}\|_2^2 \tag{2.33}$$

The updated solution based on gradient decent method is written as:

$$\begin{aligned} \mathbf{f}^{(n+1)} &= \mathbf{f}^{(n)} - \mu\nabla C(\mathbf{f}^{(n)}) \\ &= \mathbf{f}^{(n)} - \mu\sum_k \mathbf{M}_k^T\mathbf{B}_k^T\mathbf{D}^T\text{sign}(\mathbf{g}_k - \mathbf{D}\mathbf{B}_k\mathbf{M}_k\mathbf{f}^{(n)}) - \mu\lambda\nabla^T\nabla\mathbf{f}^{(n)}, \end{aligned} \tag{2.34}$$

In order to let L1 norm to be differentiable, the L1 norm is approximated as $\rho(z) = |z| \doteq \sqrt{z^2 + \epsilon^2}$, where ϵ is a very small scalar.

III. Robust function for data fidelity term, Tikhonov regularization term

Robust statistics provided M-estimator (influence function) [74] approach to SR problem. To overcome the drawbacks of L1 norm or L2 norm we talked above, a wide range of robust functions have been investigated to cut down the weights of distant points that are possible outliers. Re-descending M-estimator approach is an ideal one that can limit the influence of a distant point to some maximum value which is also called outlier threshold. Once the outlier threshold is determined, a variety of schemes can be used to eliminate the influence of outliers, either by cutting off the influence suddenly beyond the outlier threshold, or by letting the influence gradually approach to zero. The underlying assumption for Re-descending M-estimator is that the noise is basically Gaussian distributed but that outliers

may exist locally or globally across the entire image with no specific statistical model. The lack of prior knowledge on outlier limits Re-descending M-estimator since the accuracy of the influence is sensitive to the predefined outlier threshold. Extensive research has been done to decide the outlier threshold. In this section, I will investigate the case where Lorentzian function [75, 76] is used for data fidelity term, and Tikhonov regularization term is used to stabilize the SR solution. The cost function is as follows:

$$C(\mathbf{f}) = \sum_k \mathbf{1}^T \rho(\mathbf{g}_k - \mathbf{H}_k\mathbf{f}) + \lambda \|\nabla \mathbf{f}\|_2^2, \tag{2.35}$$

where $\rho_\sigma(z) = \log(1 + \frac{z^2}{2\sigma^2})$, here σ is outlier threshold. The updated solution based on gradient decent method is written as:

$$\begin{aligned} \mathbf{f}^{(n+1)} &= \mathbf{f}^{(n)} - \mu \nabla C(\mathbf{f}^{(n)}) \\ &= \mathbf{f}^{(n)} - \mu \sum_k \mathbf{M}_k^T \mathbf{B}_k^T \mathbf{D}^T \rho_\sigma^{'}(\mathbf{g}_k - \mathbf{D}\mathbf{B}_k\mathbf{M}_k\mathbf{f}^{(n)}) - \mu\lambda \nabla^T \nabla \mathbf{f}^{(n)}, \end{aligned} \tag{2.36}$$

where $\rho_\sigma^{'}(z) = \frac{2z}{2\sigma^2 + z^2}$, and different outlier threshold σ parameter settings were also proposed. Patanavijit [76] set the threshold manually. El-Yamany [75] adaptively set the outlier threshold based on similarity between the target image and warped reference image. Here, the normalized sum of absolute difference (SAD) of intensity values is calculated as similarity metric.

2.2.3 Implementation of super-resolution imaging

I develop both a C++ executable and Matlab toolbox of super-resolution imaging. In either the C++ executable or the Matlab toolbox, three kinds of super-resolution algorithms discussed in the last section are implemented. Next an example of super-resolution imaging using our developed Matlab toolbox will be shown. In this example, ten original images (320×240) are captured from a web camera; some Gaussian noise and salt pepper noise are added to the third and fifth images (shown in Figure 2.5(c) and 2.5(d)), respectively. I choose the first image as reference image(shown in Figure 2.5(a)). The enhancement factor

for super-resolution reconstruction is two. By comparison, L2 error norm (in Figure 2.6(b)) cannot eliminate outliers when reconstructing super-resolution image, on the other hand, L1 error norm (in Figure 2.6(c)) can eliminate outliers but also eliminate some useful high frequency information for SR reconstruction. A robust function (Lorentzian function is used in the experiment) can well distinguish high frequency information and outliers, resulting in sharper SR image (in Figure 2.6(d)) without artifacts as well.

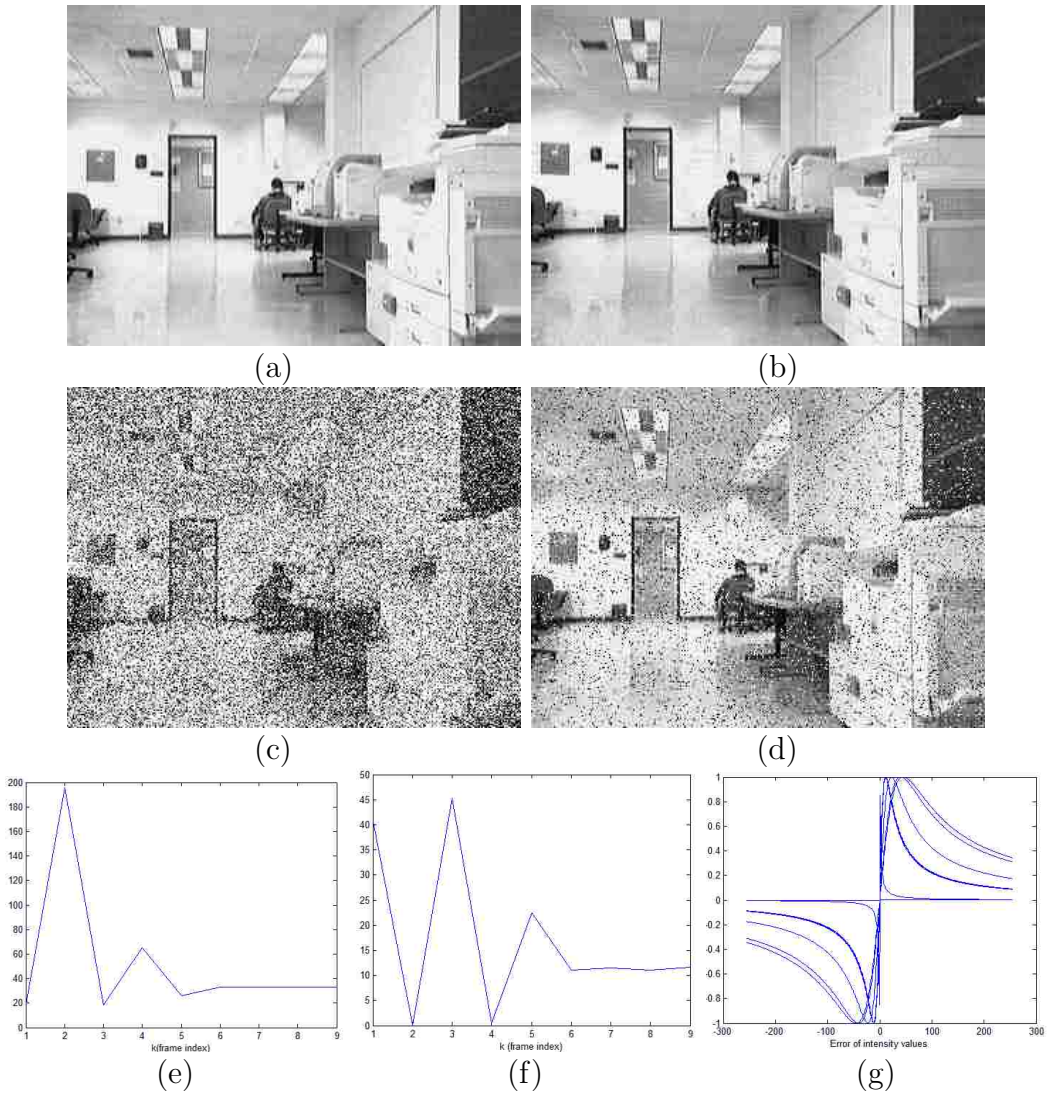


FIGURE 2.5. Input images for super-resolution. (a) and (b) Two LR images, reference and target images; (c) Gaussian noised LR image; (d) salt pepper noised LR image; (e) Sum of absolute differences (SAD) between 9 target images and warped reference image, respectively; (f) Outlier thresholds for 9 target images based on similarity metric (SAD), respectively; (g) Influence functions corresponding to 9 target images, respectively.

I use sum of absolute difference (SAD) as a similarity metric for computing outlier threshold frame by frame. Among ten original low-resolution images, the first image is used as reference image, and the other 9 images are used as target images. Among them, the second and fourth target images are degraded by Gaussian noise and salt pepper noise, respectively. I compute mean sum of absolute value for nine warped target images and reference image (in Figure 2.5(e)). The nine outlier thresholds for these nine target images are further computed based on mean SAD. From Figure 2.5(f), we can see that the larger the SAD is, the smaller the outlier threshold is. Furthermore, influence functions for each frame (in Figure 2.5(g)) with different outlier thresholds will reject pixels with low similarities as a whole. However, the similarity metric between the reference image and the warped target image has its limitation. The similarity metric depends on the texture of the image.

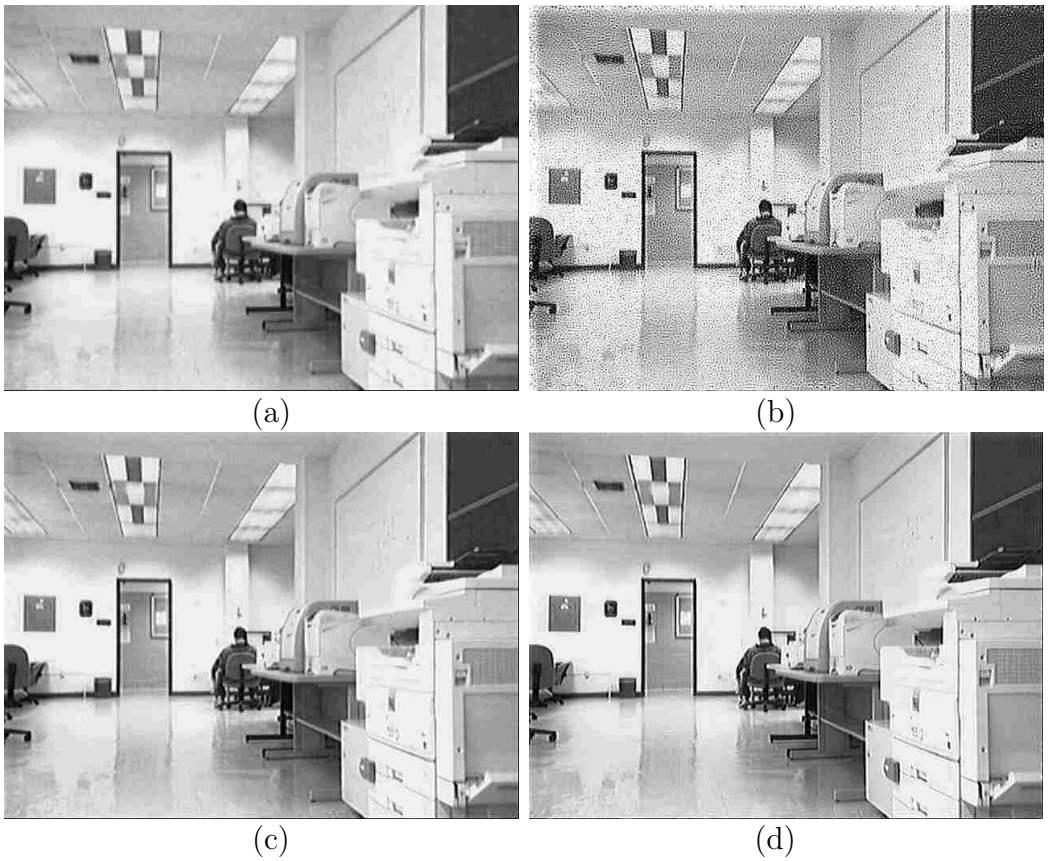


FIGURE 2.6. Super-resolution results using different error norms. (a) Bilinearly interpolated reference image; (b) Resulting SR image using L2 error norm; (c) SR image using L1 error norm; (d) SR image using robust error function.

SR reconstruction problem for image registration error is still a challenging problem. However, we may use some good image registration algorithms to reduce image registration errors in the SR problem. In my implementation, instead of using parametric or feature-based methods for image registration, I employ dense optical flow method [61] to compute motion vectors pixel wise, reducing image registration error as much as possible. I implement super-resolution using C++ as well.

In order to explore new computational photography algorithm on smartphone, I extend super-resolution reconstruction to Android phone. Apart from the above super-resolution imaging algorithmic challenges, mobile computational photography has its own challenges, including limited computational resources. Thus, in my implementation of super-resolution on smartphone, robust SR reconstruction is applied to a small part of the image because of limited memory resource for SR reconstruction. In my developed Android application, the robust SR algorithm is applied to a region of interest (ROI) to get high-resolution image. When a point is touched on screen of smartphone, a ROI is cropped from original large-size images captured back to back. In this particular application, we expected that geometric image registration would be necessary because it is almost impossible to keep a smartphone still in practice when taking multiple images. When I did experiments with a Google Nexus phone to take multiple LR images, I found that we cannot assume global parametric motion estimation even when the scene is static. This is due to the fact that a slight shake of handheld smartphone will cause an unexpected motion between photos during the capturing process. Robust and accurate registration is proposed to improve the quality of the output image. In my robust registration algorithms, median threshold bitmap (MTB) alignment is first applied to roughly get large motion between ROIs in pixel accuracy, and then region-based algorithm (optical flow method) is used to refine the motion in sub-pixel accuracy. The joint MTB alignment [77] and optical flow method [61] improves the robustness of image registration. Figure 2.7 shows result of robust SR on Android phone.

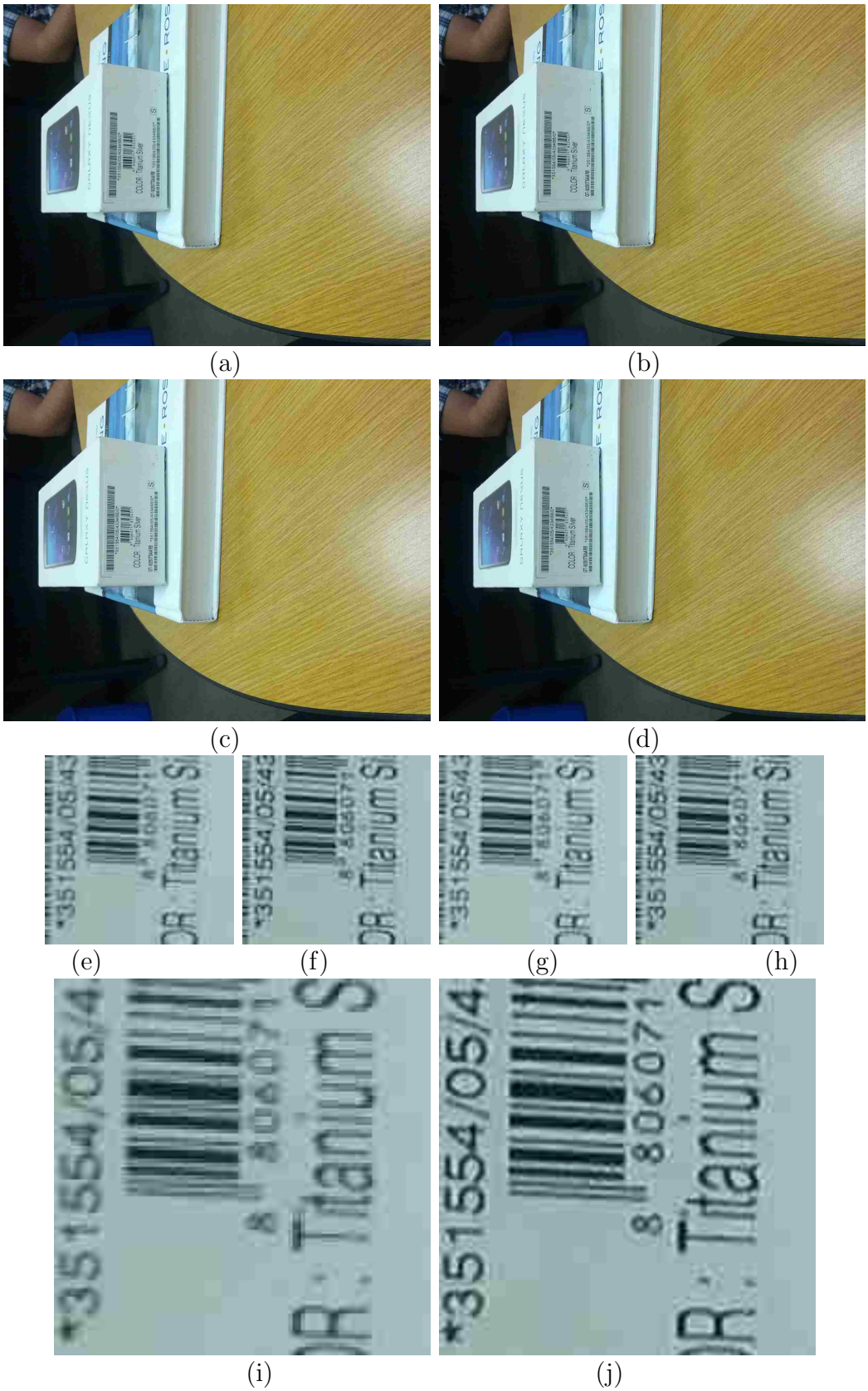


FIGURE 2.7. Super-resolution imaging on Android phone. (a)-(d) Four original LR images; (e)-(h) Four ROIs from (a)-(d), reference and three target images for super-resolution; (i) Bilinearly interpolated image of reference image (e); (j) Reconstructed super-resolution image of reference image (e).

In this example, four images are captured back to back (shown in Figure 2.7(a)-2.7(d)). A ROI is cropped from each of these four images. Here, in order to get accurate ROI, MTB alignment is used. Four ROIs (100×100) (shown in Figure 2.7(e)-2.7(h)) are merged to create a high-resolution image. By comparison of a bilinearly interpolated image (shown in Figure 2.7(i)) of the reference image and a reconstructed super-resolution image (shown in Figure 2.7(j)), the proposed robust super-resolution algorithm can be successfully applied on smartphone as new experience or application of computational photography.

2.3 High dynamic range (HDR) imaging

HDR imaging, as an emerging subfield of computational photography, aims to enhance dynamic range of a scene. Dynamic range is the range of radiance that a camera can capture. High dynamic range imaging is a technique that uses traditional cameras to capture multiple images with different exposures and then merges them to get HDR image. Due to the limitation of camera sensor, the range of values captured by a traditional camera is only two orders of magnitude, stored as a 8-bite value for each of the red, green and blue channels per pixel. However, the real world can produce a much larger range than the camera can. Thus, when taking multiple images for HDR creation, each image will try to capture a different dynamic range of the scene by varying exposure settings in the traditional camera. Alternatively, each image should at least contain a range of values representative of the scene. However, in most of current digital imaging devices, pixel values in the perceived image are nonlinearly related to the radiance values in the scene. This fact motivated multiple techniques [78, 9, 10, 11] to estimate camera response function (CRF) before merging multiple images into an HDR image. The later merged HDR image, which is also called radiance map of the scene, stores a depiction of the scene in a range of floating-point intensity values. The range of stored intensity values might be more than five orders of magnitude, which is very common in a scene simultaneously. By contrast, current display devices such as cathode ray tube (CRT),

liquid crystal displays (LCD) are able to reproduce only two orders of magnitude of intensity variations. The disparity problem necessitates the process of reducing of range of values in HDR image so that the displayable image on low-dynamic range devices is meaningful to human visual system. Extensive techniques [79, 80, 81, 82] of achieving dynamic range reduction have been proposed. These techniques are commonly called tone-mapping operators in the HDR imaging field.

In the following part, the HDR imaging will be decomposed into two main processes: HDR image creation and HDR tone mapping. HDR image creation process includes estimation of CRF and merging multiple images into an HDR image with the knowledge of CRF. HDR tone mapping is to map HDR image to a low dynamic range display device meaningful to human visual system.

2.3.1 HDR image creation

Suppose that we are given multiple images, each of which captures a limited dynamic range of the scene. However, each pixel will be suitably exposed in one or more images. Before merging multiple images into radiance map of the scene, estimation of CRF should be performed to relate the intensity values in multiple images to the radiance value in the scene. Among many CRF estimation algorithms [78, 9, 10, 11], Debevec's method [9] gains its popularity partially for its simplicity. Next Debevec's method [9] will be introduced to estimate CRF.

Assume that the camera response function $r(x)$ is given by

$$Z_{i,k} = r(L_{i,k}) = r(E_i \Delta t_k), \tag{2.37}$$

where $Z_{i,k}$ denotes intensity value at ith pixel location in kth image, Δt_k is exposure duration for kth image, $L_{i,k}$ is recorded radiance value, and E_i is irradiance value. We denote the logarithmic response function $s(Z_{i,k})$ as $s(Z_{i,k}) = \ln(r^{-1}(Z_{i,k}))$, by taking logarithmic operation on both sides of camera response function, we will get

$$s(Z_{i,k}) = \ln(E_i) + \ln(\Delta t_k). \tag{2.38}$$

To derive the logarithmic response function $s(Z_{i,k})$, the cost function to be minimized is as follows:

$$C = \sum_i \sum_k w(Z_{i,k})[s(Z_{i,k}) - \ln(E_i) - \ln(\Delta t_k)] + \lambda \sum w(z)s''(z), \tag{2.39}$$

The cost function consists of data fidelity term and regularization term which uses second derivative to smooth the derived response function. In order to give more weights to pixel around the middle range of intensity values, the weighting function $w(Z_{i,k})$ chooses a simple hat function:

$$w(z) = \begin{cases} z - Z_{min} & \text{if z} \leq 1/2(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{if z} \geq 1/2(Z_{min} + Z_{max}) \end{cases} \tag{2.40}$$

where Z_{min} and Z_{max} are intensity values for saturation regions. The cost function is minimized using the singular value decomposition method. The CRFs for RGB channels will be estimated separately. With the knowledge of CRF which is estimated above, an HDR image is then computed from N exposures by the following equation:

$$E_i = \sum_k \frac{r^{-1}(Z_{i,k})w(Z_{i,k})}{\Delta t_k} / \sum_k w(Z_{i,k}). \tag{2.41}$$

If the camera and the scene are static during the entire imaging process, HDR imaging is relatively straightforward. However, the camera may shift slightly, which results in misalignments between multiple images. In addition, there may also exist moving objects in the scene which also degrade the result. There are several published techniques addressing the image registration problem in HDR creation. Among them, the two popular algorithms are the Kang's method [83] and Ward's method [77] (Median Threshold Bitmap (MTB)). Kang's method [83] computes motion vector for each pixel between successive frames. MTB alignment [77] performs image alignment on bitmaps rather 8-bit grayscale images, also, this method does not perform image warping either. Thus, MTB alignment is ten times faster than Kang's method. However, MTB alignment does not address the problem of arbitrary camera movement or object motion in the scene. By comparison, MTB alignment is suitable

for slight camera movement between successive frames. On the other hand, Kang's method is more suitable for HDR video where significant motion is expected in the video frames.

2.3.2 HDR tone mapping

Once the radiance map of the scene is estimated, a tone mapping operation is in general needed to display the recorded radiance on low dynamic range devices. There are two main approaches for tone mapping: global tone mapping operators [79], which are very efficient in terms of computational complexity; and local tone mapping operators [80, 81, 82], with typically better results but at the expense of higher computational cost. Next a typical global tone mapping operator proposed by Reinhard [79] will be investigated.

The tone mapping operator is performed on luminance channel of the scene which is obtained from the input red R, green G and blue B channels with $Y = 0.21R+0.72G+0.07B$. First, the input luminance map is linearly scaled to the estimated key of the scene. The key of the scene is a number which indicates how bright or dark the overall scene is. Reinhard [79] uses log average luminance across the whole image as estimation of key of the scene as follows:

$$Y_{av} = \exp(\frac{1}{N}\sum_{i}\log(\delta + Y_i)), \tag{2.42}$$

where N is the number of pixels in the input image, and δ is a small scalar number to avoid singularity of log operation on possible pixels. The policy of scaling is that, Reinhard [79] would like log average luminance of average-key scene to be mapped to 18 percentage of the display range which is common practice in photography. The higher the key of scene is, the higher the mapped value is. The linear scaling is given by:

$$\hat{Y_i} = \frac{a}{Y_{av}}Y_i, \tag{2.43}$$

where a is a given user parameter which is 0.18 by default. However, the above linear scaling does not expect the mapped value in the display range, the following compressive function

is performed such that the scaled value is within the range of [0, 1]:

$$Y_{d,i} = \frac{\hat{Y}_i}{1 + \hat{Y}_i}. \tag{2.44}$$

The above compressive function scales small numbers linearly, while higher luminance values are largely compressed. In order to let the user control the smallest luminance value to be mapped to white, the above compressive function is modified to the following transfer function:

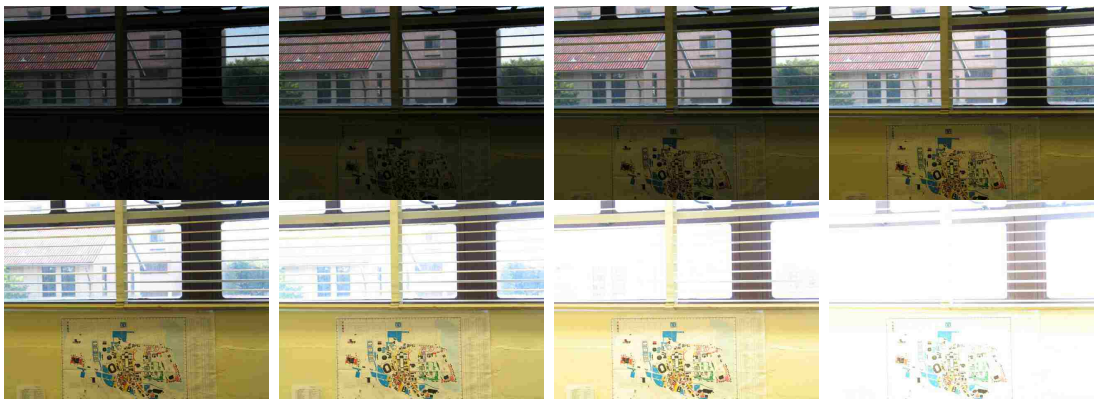
$$Y_{d,i} = \frac{\hat{Y}_i(1 + \frac{\hat{Y}_i}{Y^2_{white}})}{1 + \hat{Y}_i}, \tag{2.45}$$

where Y_{white} is user parameter by default which is set as maximum luminance value across the whole image after linear scaling. This global tone mapping operator can be modified to become local tone mapping operator as well. However, the readers are encouraged to refer to the original paper [79].

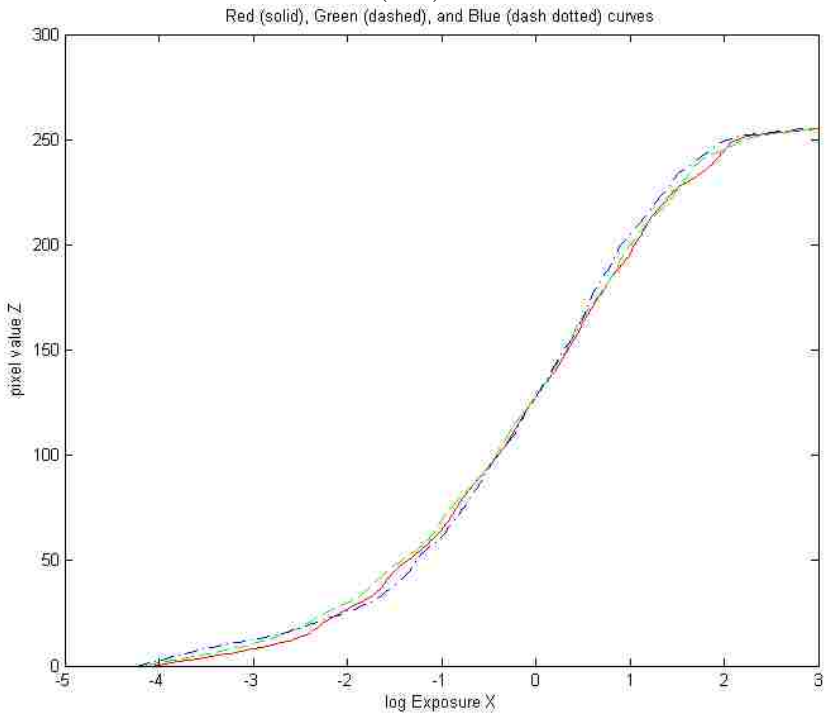
One of typical local tone mapping operators is proposed by Durand [82]. In this method, it decomposes the image into two layers: base layer and details layer. The contrast reduction is performed only on base layer which encodes the large-scale variations. Thus this tone mapping operator greatly reduces contrast while preserving details. Particularly, the base layer is obtained by using bilateral filtering technique which employs weighting function considering both spatial domain and intensity domain. The weighting function puts fewer weights to pixels with large intensity differences. This preserves strong edges after filtering input image.

2.3.3 An example of HDR imaging

I implement HDR imaging on PC using Matlab and on Android phone using Java. An example of Matlab implementation is shown is Figure 2.8. In this example, eight photographs are taken at 1-stop increments from 1/640 sec to 1/5 sec. These eight images are well aligned so no image registration algorithm is needed. The camera response functions for the red, green and blue channels are shown in Figure 2.8(i).



(a-h)



(i)

(j) (k)

FIGURE 2.8. High dynamic range image creation. (a-h) Eight photographs taken at 1-stop increments from 1/640 sec to 1/5 sec; (i) The recovered camera response function for red, green and blur plotted on the same axes; (j) Tone mapped HDR image using Reinhard's method; (k) Tone mapped HDR image using Durand's method.

After we estimate HDR image, both Reinhard's global tone mapping operator and Durand's local operator are used to make the tone mapped HDR image displayable on the LDR display device. We can see that from the two tone mapped images, the local operator can preserve more details than the global operator.

2.4 Focus stacking

Depth of field is defined as the distance between nearest and farthest objects that appear sharp in the perceived image. Factors affecting depth of field include circle of confusion which is informally defined as the largest circle that is indistinguishable from a point, focal length and the distance from object to camera. Many computational cameras have been designed to extend depth of field. Dowski [6] used wavefront coding that regardless of depth, the object will be in focus (small PSF) for some parts of the lens or blurry (large PSF) for other parts of the lens. Levin [7] designed coded aperture to recover the latent image based on the point spread function of blur and depth as well. Cossairt [84] added a random diffuser to make the PSF stochastic, and ultimately less dependent on ray angles, leading to depth invariance. Another approach is to extend the depth of field through pure image processing techniques without the knowledge of camera design or point spread function of blur which causes defocus. Focus stacking is one such technique that is used to extend depth of field by taking multiple pictures that are focused at different depths. If the scene and the camera are both static during the imaging process, then there is no need to do geometric registration and the process of focus stacking becomes relatively easy. Many methods [19, 20, 21, 22, 23, 24] have been proposed to extend the DOF by means of first detecting in-focus regions from images with different focus areas and then constructing an all-in-focus image by selection of pixel detected as in-focus. The earliest method may be dated back to Pieper and Korpel [19], they proposed to use point-based image fusion in which finite difference filtering is employed to detect in-focus regions and a maximum selection rule is used to select the pixel in focus.

The latter variance method assumes that in-focus regions have larger intensity variations over a small window. The pixel with the highest variance at the same position is selected in the fused image, similar to point-based method, this method is claimed to be computationally simple [20]. In addition, wavelet-based focus stacking [21] computes the wavelets transform of each image and selects maximum-absolute-value of wavelet coefficients which corresponds to sharper brightness changes to build up the wavelet coefficients of fused image. This method leads to better results but generally at a cost of larger memory occupation and computation time. Next I will discuss the local sharpness method for focus stacking. The use of local sharpness as an indicator of focus is a commonly used idea in focus stacking. The local sharpness $\widetilde{w}_i(\mathbf{x})$ at pixel $\mathbf{x}$ of image $\widetilde{\mathbf{f}}_i$ (where $\widetilde{\mathbf{f}}_i$ is aligned image)is defined as

$$\widetilde{w}_i(\mathbf{x}) = \sum_{\mathbf{y} \in N_h(\mathbf{x})} \|\nabla \widetilde{\mathbf{f}}_i(\mathbf{y})\|, \tag{2.46}$$

where $\nabla \widetilde{\mathbf{f}}_i(\mathbf{y})$ is the gradient vector at pixel $\mathbf{y}$ obtained by applying a Sobel filter, $\|\cdot\|$ denotes the gradient magnitude, and $N_h(\mathbf{x})$ is a $h \times h$ window around pixel $\mathbf{x}$. The calculated weight maps need to normalized:

$$w_i(\mathbf{x}) = \frac{\widetilde{w}_i(\mathbf{x})}{\sum_{i=1}^{N} \widetilde{w}_i(\mathbf{x})}, \tag{2.47}$$

When all weights are zero, then division by zero occurs in the above normalization. $1/N$ is assigned to each weight to avoid singularity. The image fusion to obtain the all-in-focus image $\mathbf{f}$ is then performed according to the following equation:

$$\mathbf{f}(\mathbf{x}) = \sum_i \widetilde{\mathbf{f}}_i(\mathbf{x}) w_i(\mathbf{x}), \tag{2.48}$$

Next I will give an example of the focus stacking method (shown in Figures 2.9 and 2.10) for extending depth of field. In this example, two images (shown in Figures 2.9(a) and 2.9(b)) are aligned beforehand. The weighting maps based on local sharpness method are shown in Figures 2.9(c) and 2.9(d). The created all-in-focus image is shown in Figure 2.9(f), I also create the average of two input images shown in Figure 2.9(e). By comparison (shown in

Figure 2.10), the local sharpness method can successfully detect in-focus regions and the image fusion based on weighting maps can extend depth of field of the scene without the knowledge of PSF of defocus blur or camera aperture design pattern.

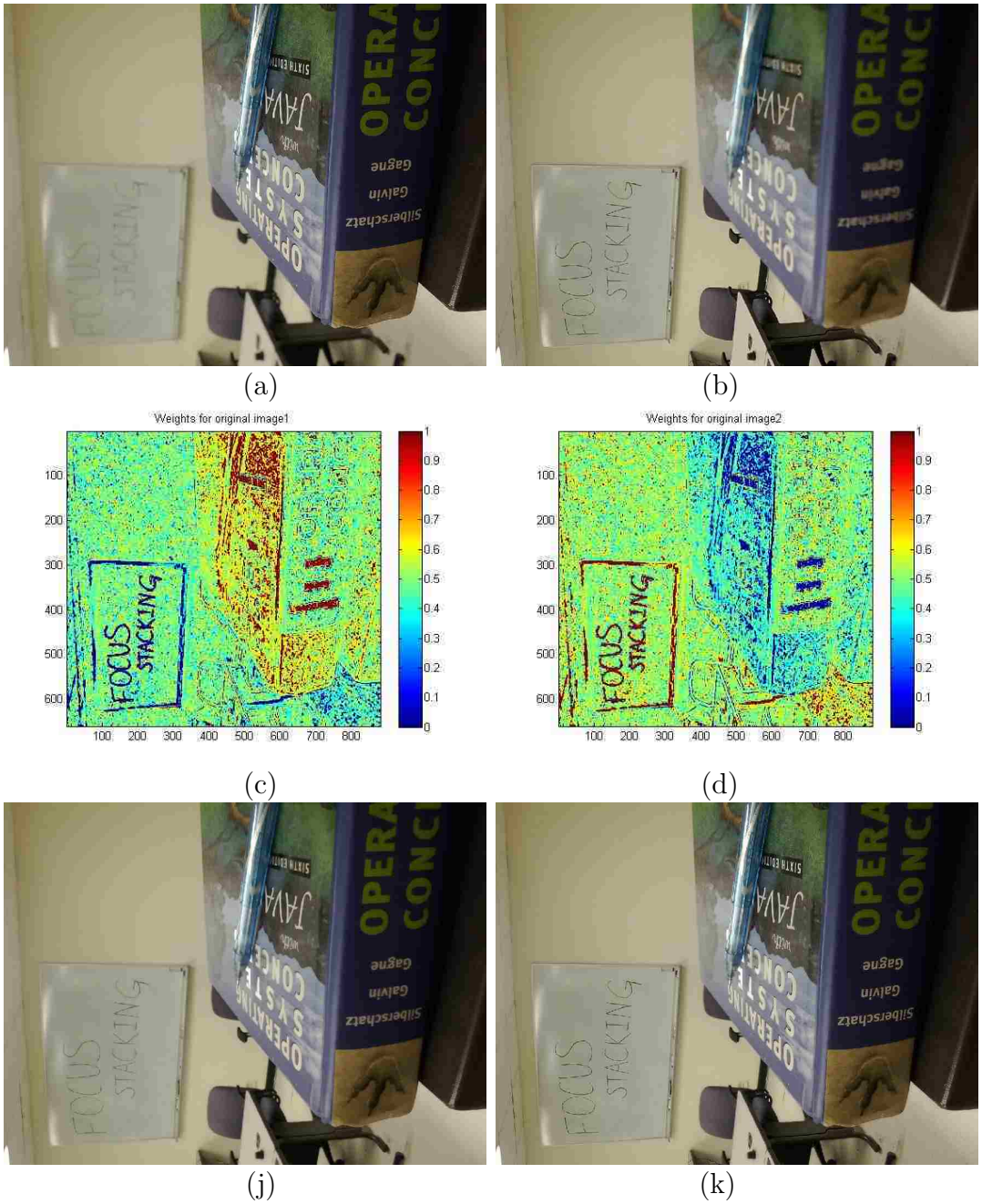


FIGURE 2.9. Focus stacking from multiple images. (a-b) Two images with different in-focus regions; (c-d) The weighting maps based-on local sharpness criteria; (e) The average image of (a) and (b) ; (f) Focus stacked image based on weighting maps.

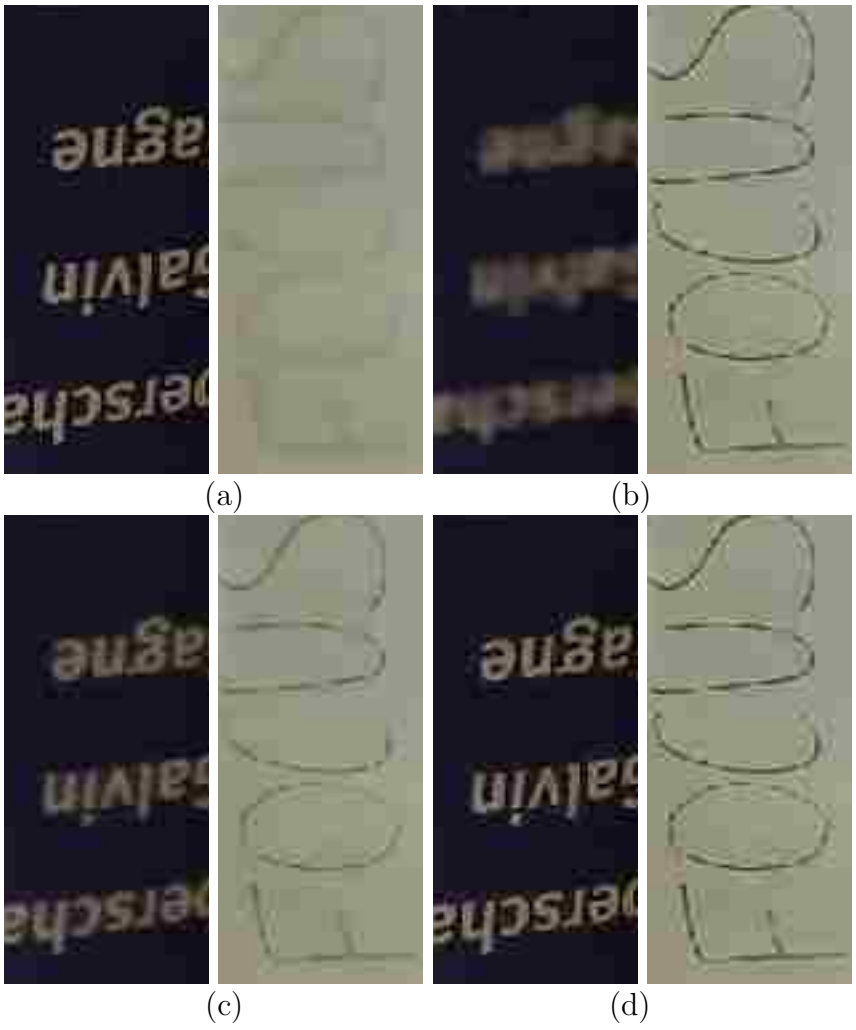


FIGURE 2.10. Zoomed-in focus stacking from multiple images. (a) and (b) Two images with different in-focus regions;; (c) The average image of (a) and (b); (d) Focus stacked image based on weighting maps..

3 Super-resolution restoration of motion blurred images

3.1 Introduction

Super-resolution (SR) image restoration [13, 14, 15, 16, 17, 18] has been extensively studied over the last two decades, however there are still open problems and challenges that need to be researched further. One of the most important challenges in multi-frame super-resolution restoration has been the motion estimation problem. Accurate motion field is essential in the success of SR restoration; many super-resolution papers report results with data where the motion field is restricted to a parametric model, such as perspective model or even translational model. This is obviously very limiting and may not be used in most real-life applications. To deal with more realistic motion, block-based motion estimation or optical flow methods [59, 60] could be utilized. In recent years, highly accurate and practical optical flow methods [16, 61] have been proposed; and these optical flow methods definitely improve the performance in super-resolution applications. A second challenge in super-resolution restoration is blur kernel estimation. The majority of existing super-resolution algorithms assume identical blur kernel for all input images; the blur kernel is typically modeled as a symmetric Gaussian function, whose standard deviation is estimated empirically or by some parametric estimation method. The assumption of identical blur kernel does not hold when there are fast moving objects in the scene or the camera is shaken during the exposure time. Ideally, we should estimate the blur kernel for each input image separately. There are few super-resolution methods that estimate blur kernel for each image. In [65], a region based

matching is first used to track the moving object of interest in the image sequence and then the motion blur direction and magnitude are estimated from tracked displacements. The motion field is limited to affine model, and so is the motion blur kernel. In [64], a Bayesian approach is proposed for adaptive super resolution that incorporates high-resolution image restoration, optical flow, noise level and blur kernel estimation. The estimation process is reduced to an individual component given the other terms; and the Bayesian inference iterates four steps: optical flow, noise estimation, blur estimation and image restoration. The drawback of the method is that the blur kernels are limited to a Gaussian function with possibly different standard deviations. In [85], the motion field is used to construct the motion blur for each frame; however, it is obvious that such an approach cannot account for intra-frame motion blur.

In this chapter, I will investigate super-resolution restoration of images which are possibly degraded with large motion blur. I do not make any restriction on the exposure time or the amount of motion: The blur kernel for each input image is estimated separately; therefore, intra-frame motion blur is allowed. To achieve this, I will turn to the recent developments in blind deconvolution. One group of successful single-image motion deblurring methods incorporates various sparse priors for either the latent images or motion blur kernels [33, 34, 35]. Another group of successful methods tries to explicitly detect salient image structures for kernel estimation [37, 36]. A two-phase kernel estimation algorithm is recently presented in [38], where quick kernel initialization based on strong edge selection, kernel refinement and latent image restoration based on TV-L1 optimization are separately addressed. Our initial motion deblurring step is based on an improved version of Cho's method [37].

After the motion deblurring step, motion field will be estimated without any restriction to a parametric model. Finally, super-resolution restoration will be applied to the motion deblurred images. The first two steps, motion blur estimation and motion field estimation, may have estimation errors, which would result in artifacts in the final SR image. Therefore,

I propose a weighted cost function for the super-resolution restoration step, where a weight associated with an input image reflects the reliability of the corresponding kernel estimate and the deblurred image.

This chapter is organized as follows. In Section 3.2, I will detail the proposed method, which includes three main steps: (1) single image motion deblurring for motion blurred images, (2) motion estimation, and (3) super-resolution restoration. In Section 3.3, experimental results from real video data will be provided. I will conclude in Section 3.4.

3.2 The proposed framework

In my framework, the image formation process is modeled as a two-step process. In the first step, low-resolution images $\{\mathbf{g}_i\}$, $i = 1, ..., N$, are generated from a high-resolution image $\mathbf{f}$ through a sequence of linear operations: image warping $\{\mathbf{M}_i\}$, image blurring $\mathbf{B}$ with a common space-invariant point spread function (PSF), and downsampling $\mathbf{D}$. This first step of the imaging process can be formulated as

$$\mathbf{g}_i = \mathbf{DBM}_i\mathbf{f}. \tag{3.1}$$

In the second step, a single-image blur process is formulated as:

$$\widetilde{\mathbf{g}}_i = \mathbf{H}_i\mathbf{g}_i, \tag{3.2}$$

where $\mathbf{g}_i$ is the latent image, $\mathbf{H}_i$ is the matrix associated with the blur kernel, and $\widetilde{\mathbf{g}}_i$ is the blurred image.

The reason I separate the imaging process into two steps is that large motion blur will be dealt with and therefore the motion field from the input images $\{\widetilde{\mathbf{g}}_i\}$ cannot be reliably estimated. I first need to deblur $\{\widetilde{\mathbf{g}}_i\}$ to obtain the latent images $\{\mathbf{g}_i\}$, and then estimate motion field, which will then be used in the super-resolution step.

Because the estimated blur kernels and therefore the deblurred images $\{\mathbf{g}_i\}$ may not be accurate, I will use a weighted cost function for the super-resolution restoration step, where

a weight w_i associated with an input image reflects the reliability of the corresponding kernel estimate and the deblurred image.

The framework to achieve super-resolution of motion blurred images is given in Figure 3.1. The motion deblurring step estimates the motion blur kernels $\{\mathbf{H}_i\}$ and the latent (deblurred) images $\{\mathbf{g}_i\}$ from the input images $\{\widetilde{\mathbf{g}}_i\}$. The weights $\{w_i\}$, reflecting the reliability of motion deblurring, are estimated. The optical flow step estimates the motion vectors $\{\mathbf{M}_i\}$ between the reference image $\mathbf{g}_r$ which is selected from N deblurred images and the input images $\{\mathbf{g}_i\}$. In the last step, I apply weighting-based super-resolution where motion deblurred images $\{\mathbf{g}_i\}$, estimated motion vectors $\{\mathbf{M}_i\}$ and calculated weights $\{w_i\}$ are taken as input, with $\mathbf{g}_r$ set as the reference image. The super-resolution image is denoted as $\mathbf{f}$. I now discuss the details of the framework.

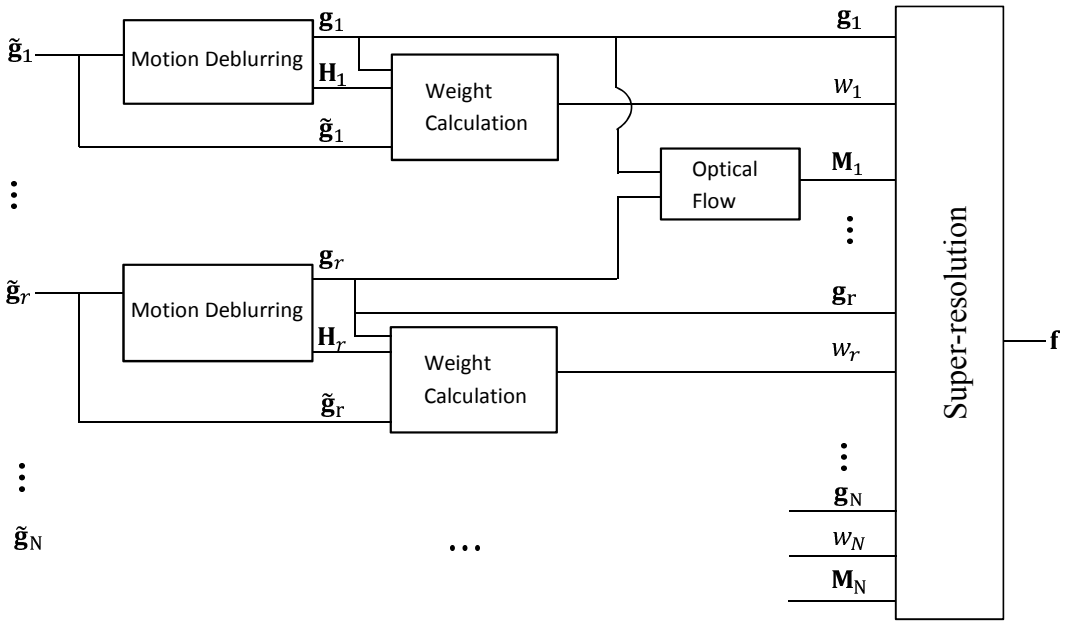


FIGURE 3.1. The framework for super-resolution of motion blurred images.

3.2.1 Single image motion deblurring using TV-L1 model

My single image deblurring algorithm consists of three steps. First, the blur kernel is initialized using the kernel initialization method presented by Cho [37]. Second, the kernel is refined

through hard thresholding [34] by setting kernel values that are less than the 10 percent of the maximum value in the kernel to 0. While Xu [38] also presented a kernel refinement step, it is computationally costly and did not improve significantly over the simple hard thresholding method mentioned. Finally, the latent image is estimated from the estimated blur kernel and original blurred image. It is well-known that L2 norm in the data fidelity term is not effective to avoid outliers; in addition, L2 norm in the regularization term also results in over-smooth restoration. To achieve robustness to outliers as well as sharp results, I used the TV-L1 based method [55], which minimizes the energy function

$$E(\mathbf{g}_i) = \|\widetilde{\mathbf{g}}_i - \mathbf{H}_i\mathbf{g}_i\|_1 + \lambda\|\nabla\mathbf{g}_i\|_2, \tag{3.3}$$

where $\mathbf{H}_i$ is the estimated blur kernel, $\widetilde{\mathbf{g}}_i$ is the original blurred image, $\mathbf{g}_i$ is the latent image to be estimated, and λ is the regularization parameter. Clearly, it brings non-linearity and non-differentiability to both data and regularization terms, resulting in computational difficulty to solve this problem. To overcome the computational difficulties, Wang [55] proposed an alternating minimization method based on half quadratic splitting to solve the TV-L1 optimization problem. I followed the exact algorithm [55] to achieve motion deblurring. In my implementation, the regularization parameter λ is set to be 10.

3.2.2 Optical flow method for motion estimation between deblurred images

Given the motion deblurred images, the motion vectors are estimated. The reason motion estimation is done with deblurred images is that motion blur may degrade the accuracy of motion estimation significantly, causing undesired artifacts in the restored image. As shown in Figure 3.1, $\mathbf{g}_r$ is chosen as the reference image and $\{\mathbf{g}_i\}$ are chosen as target images for SR restoration. I first up-sample both the reference image and target images to the desired high-resolution lattice, and then estimate the motion fields between the upsampled reference image and the upsampled target images using the optical flow method proposed by Liu [61]. This optical flow method is based on a robust data term and discontinuity-preserving total-

variation regularization and results in accurate dense flow fields. Using this method, I obtain the motion fields $\{\mathbf{M}_i\}$ between the reference image $\mathbf{g}_r$ and the target images $\{\mathbf{g}_i\}$. (Note that $\mathbf{M}_r = \mathbf{0}$.)

3.2.3 Weighting-based super-resolution (SR) restoration method

The standard SR algorithm with Tikhonov regularization minimizes the following cost function:

$$C(\mathbf{f}) = \sum_i \|\mathbf{g}_i - \mathbf{DBM}_i\mathbf{f}\|_2^2 + \lambda\|\nabla\mathbf{f}\|_2^2 \tag{3.4}$$

where λ is regularization parameter. In order to be able to deal with estimation errors from the motion deblurring step, I modify this cost function to include the reliability of each motion deblurred image:

$$C(\mathbf{f}) = \sum_i w_i\|\mathbf{g}_i - \mathbf{DBM}_i\mathbf{f}\|_2^2 + \lambda\|\nabla\mathbf{f}\|_2^2 \tag{3.5}$$

where w_i is the weight associated with the motion deblurred image, indicating the contribution of the corresponding motion deblurred image in the overall cost to be minimized. It is reasonable to assign more weights to reliable motion deblurred images, whereas less weights to unreliable motion deblurred images.

Each weight will be calculated using the reconstruction error $\|\widetilde{\mathbf{g}}_i - \mathbf{H}_i\mathbf{g}_i\|_2$. A small reconstruction error indicates accurate blur kernel estimate and deblurred image; while a large reconstruction error indicates less reliable kernel estimate and deblurred image. The weight should then be inversely proportional to the reconstruction. The un-normalized weight $\widetilde{w}_i$ is defined as

$$\widetilde{w}_i = \frac{1}{\|\widetilde{\mathbf{g}}_i - \mathbf{H}_i\mathbf{g}_i\|_2 + \epsilon} \tag{3.6}$$

where ϵ is a small scalar to avoid singularity. Since we have more confidence in the well deblurred images than poorly deblurred images, we need to normalize the weights and assign more weights to well deblurred images than to poorly deblurred images, the normalization

is defined as

$$w_i = (\widetilde{w}_i / \max\{\widetilde{w}_i\})^4 , \tag{3.7}$$

where the fourth power is chosen empirically. The objective function in Equation 3.5 is a weighted least squares problem with Tikhonov regularization. Due to the pixel-wise warping operation in the imaging process, a closed form solution is not possible; however, the optimization can be achieved iteratively using the gradient descent technique. Starting with an initial estimate $\mathbf{f}^{(0)}$, the updated estimate at the nth iteration is

$$\begin{aligned} \mathbf{f}^{(n+1)} &= \mathbf{f}^{(n)} - \mu \nabla C(\mathbf{f}^{(n)}) \\ &= \mathbf{f}^{(n)} - \mu \sum_i w_i \mathbf{M}_i^T \mathbf{B}^T \mathbf{D}^T (\mathbf{g}_i - \mathbf{DBM}_i \mathbf{f}^{(n)}) - \mu\lambda \nabla^T (\nabla \mathbf{f}^{(n)}), \end{aligned} \tag{3.8}$$

where μ is the step size, which we empirically set to 0.1 in my experiment. Also, in my experiment, I empirically set regularization parameter λ to be 0.2. The stopping condition for gradient descent method is predetermined number of iterations, which is set to be 80 in our experiment.

While $\{\mathbf{H}_i\}$ are estimated for all input images and incorporate large motion blur as well as other blur types such as out-of-focus and sensor blurs, in my model I also have the blur $\mathbf{B}$ which is common to all images. The simplest choice for this blur is identity matrix, which corresponds to a blur kernel of delta function. Instead of a delta function, a small blur kernel is also possible which would model spatial averaging before downsampling. (In my implementation, I used a 3×3 Gaussian kernel with standard deviation equal to 1.)

3.3 Experimental results

I conduct an experiment to examine the performance of the proposed super-resolution method under unknown motion blur kernels and arbitrary mutual motions. In the experiment, a video sequence is captured with a hand-held camera and six of the video frames are used to reconstruct a super-resolution image. The input images are shown in Figures

3.2(a1)-3.2(a6). The estimated blur kernels corresponding to these images are given in Figures 3.2(b1)-3.2(b6), respectively; and the motion deblurred images are given in Figures 3.2(c1)-3.2(c6). It is clear that the motion deblurring worked well for some input images, such as the one in Figure 3.2(a5), but not as well for some others, such as the one in Figure 3.2(a4). The weights $\{w_i\}$ are 0.23, 0.67, 0.14, 0.15, 1.0 and 0.74, respectively; and they indeed reflect the quality of motion deblurring as can be seen from Figures 3.2(c1)-3.2(c6).

FIGURE 3.2. Motion deblurring. (a1)-(a6) Input images, with size 320×240. (b1)-(b6) Estimated blur kernels, with size 19×19. (c1)-(c6) Motion deblurred images.

To do super-resolution restoration, first, the motion fields need to be estimated. The first input image is set as the reference; and Figures 3.3(b)-3.3(f) show the estimated motion fields from the motion deblurred images in Figures 3.2(c2)-3.2(c6) to the reference image in Figure 3.2(c1). Next, super-resolution restoration is applied to improve the resolution by a factor of two (in Figure 3.4).

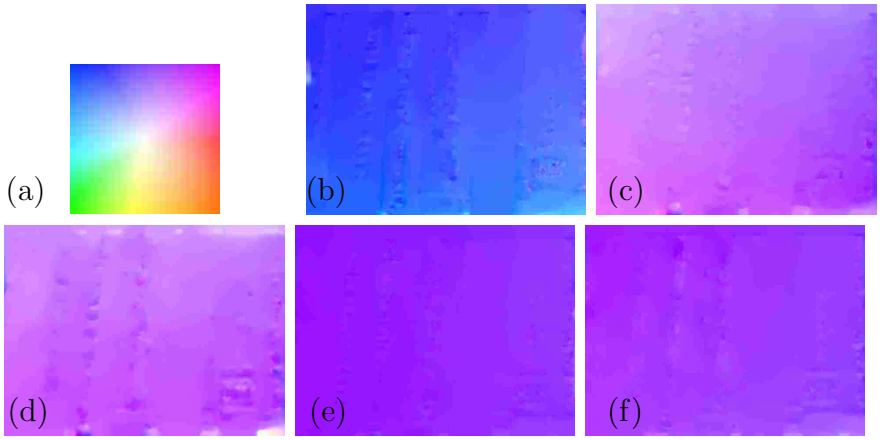


FIGURE 3.3. Motion estimation. (a) Color coding of motion vectors. (b-e) Estimated flow fields between the reference image in Figure 2(c1) and the target images in Figures 2(c2)-2(c6).

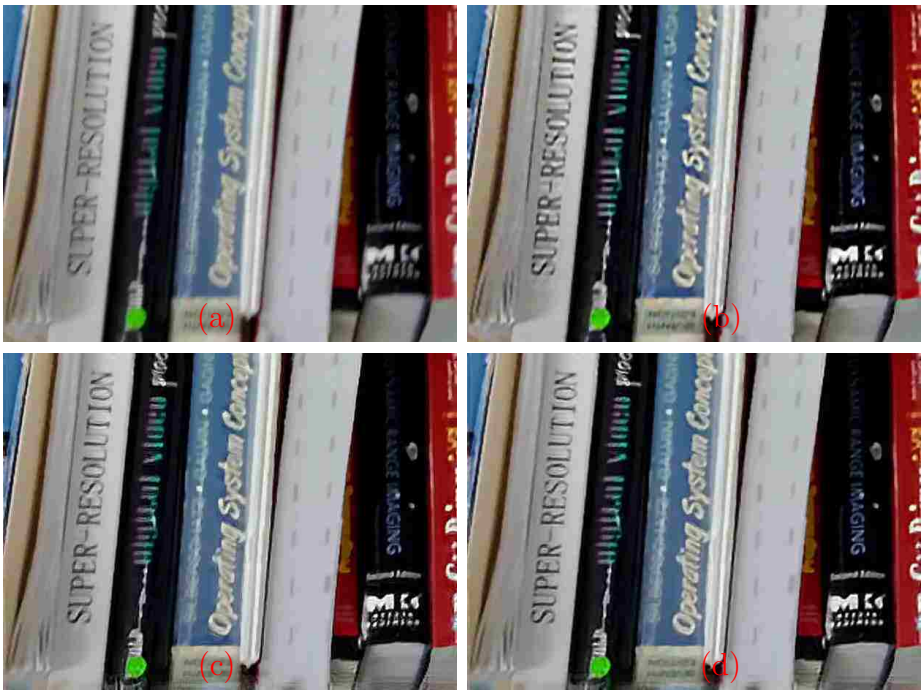


FIGURE 3.4. Super-resolution when the first input image is set as the reference. (a) Bilinearly interpolated input image; (b) Bilinearly interpolated deblurred image; (c) Image restored using standard SR without weights; (d) Image restored using the proposed SR with weights.

Figure 3.4(a) shows the bilinearly interpolated reference image; and Figure 3.4(b) shows its motion deblurred version. Figure 3.4(c) shows the result with standard super-resolution without any weights; Figure 3.4(d) shows the result with the proposed super-resolution algorithm where the weights are included. While no significant difference between the standard and proposed algorithms is visible in Figure 3.4; we can observe that the proposed algorithm has less artifacts than the standard algorithm in the zoomed-in regions given in Figure 3.5.

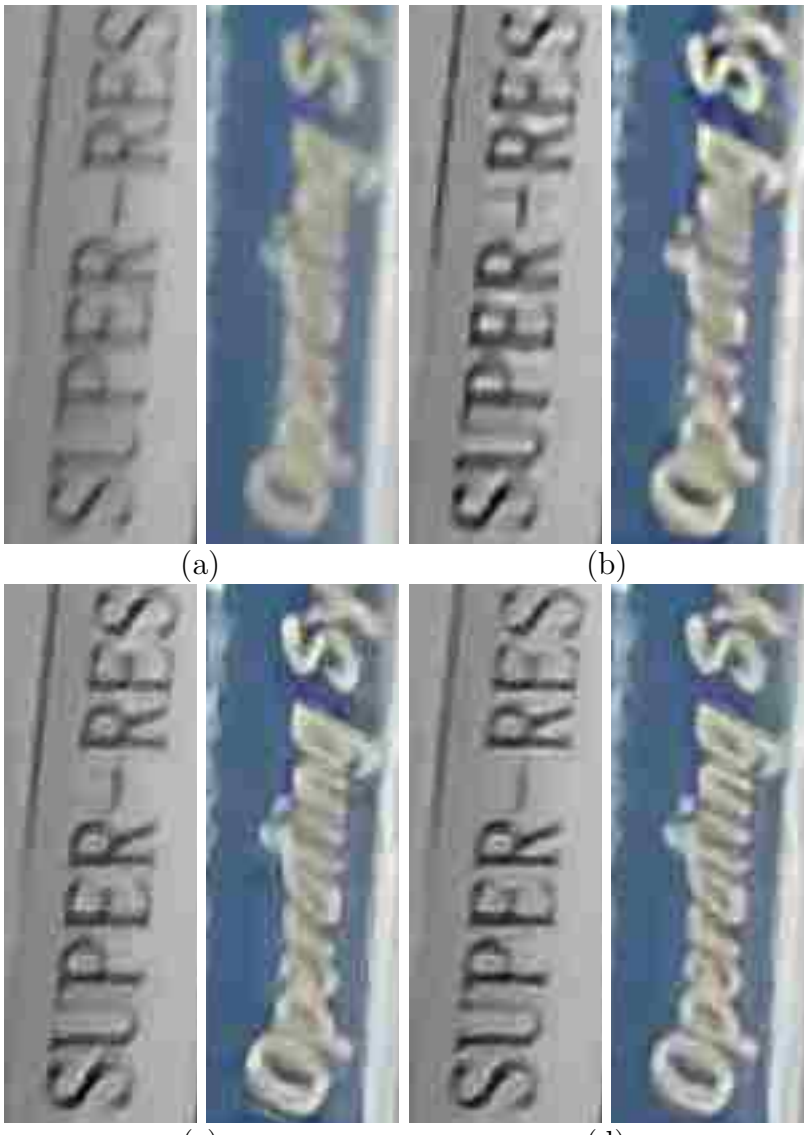


FIGURE 3.5. Super-resolution when the first input image is set as the reference. (a) Bilinearly interpolated input image; (b) Bilinearly interpolated deblurred image; (c) Image restored using standard SR without weights; (d) Image restored using the proposed SR with weights.

The effect of including weights becomes dramatic when the initial estimate is very poor. Figure 3.6 provides results when the fourth input image, given in Figure 3.2(a4) is set as the reference. As seen in the zoomed-in regions (Figure 3.7), the proposed algorithm provides much better result than the standard algorithm.

FIGURE 3.6. Super-resolution when the fourth input image is set as the reference. (a) Bilinearly interpolated input image; (b) Bilinearly interpolated deblurred image; (c) Image restored using standard SR without weights; (d) Image restored using the proposed SR with weights.

3.4 Conclusions

In this chapter super-resolution restoration from motion blurred images is investigated. In addition to a specific framework for handling motion-blurred images, a weighting-based super-resolution method is proposed which assigns more weights to well deblurred images and less weights to poorly deblurred images, thus achieving robustness to motion deblurring errors. The proposed framework contains three main steps: (1) single-image motion deblurring, (2) motion estimation using deblurred images, and (3) weighting-based method for solving the super-resolution problem. The experimental results suggest that the weighting scheme re-

duces artifacts due to errors in blur kernel estimation and motion deblurring. One limitation of the proposed algorithm is the assumption of space-invariant blur; thus incorporation of space-varying blur estimation should be investigated in my future work.

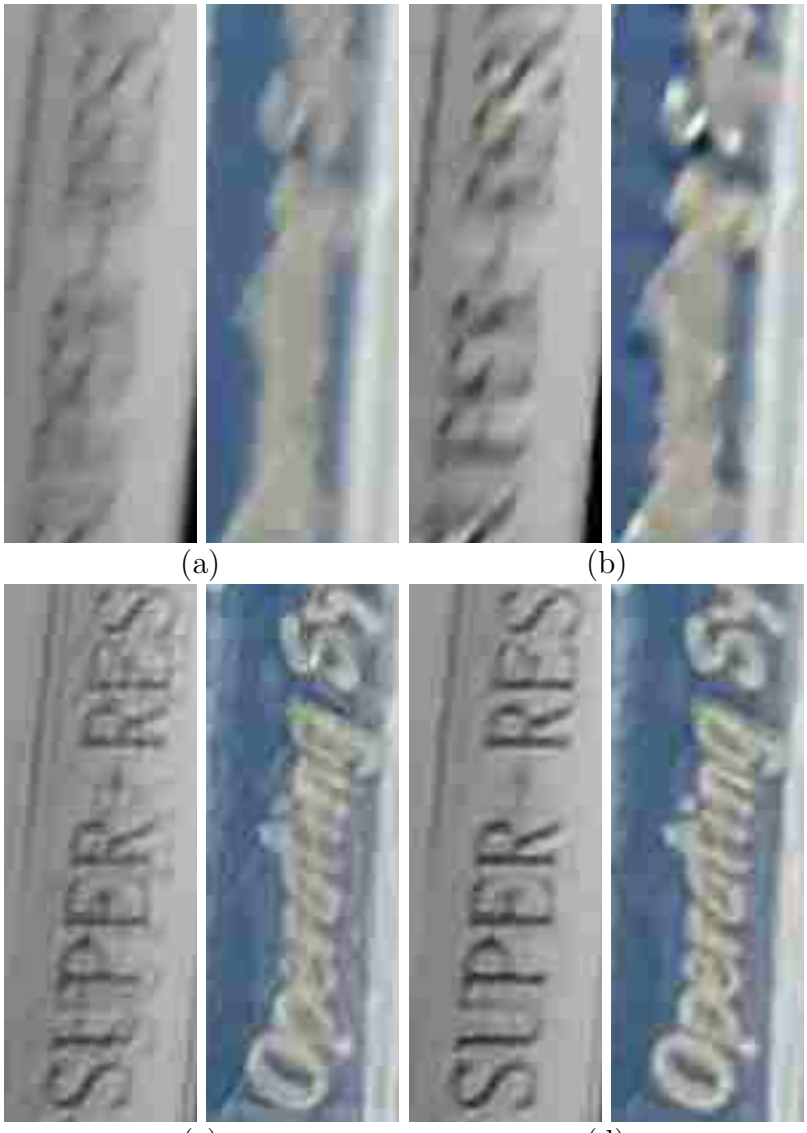


FIGURE 3.7. Super-resolution when the fourth input image is set as the reference. (a) Bilinearly interpolated input image; (b) Bilinearly interpolated deblurred image; (c) Image restored using standard SR without weights; (d) Image restored using the proposed SR with weights.

4 Space-varying blur kernel estimation and image deblurring

4.1 Introduction

In this chapter, I will investigate the image deblurring problem from a single image degraded by space-varying blur across the entire image. In such situations, the blur kernel cannot be modeled as a uniform kernel across the whole image as presented in most existing blind image deblurring problems. As discussed in the single image deblurring problems in Chapter 2, a particular blur type, including defocus blur, motion blur by camera shake or motion blur by moving object should be assumed beforehand. However, in this chapter, I do not make any restrictions on the type of blur or how the blur varies spatially. That is, the blur might be, for instance, a large (non-parametric) motion blur in one part of an image and a small defocus blur in another part without any smooth transition. The proposed space-varying image deblurring algorithm relies on the observation that the blur kernel is uniform in a local region with rich texture. Based on this assumption, I propose a three-step framework: (1) Coarse PSF estimation, (2) PSF refinement through kernel clustering, (3) Space-varying deblurring through image fusion. Regarding the PSF estimation in the first two steps, the main difficulty is that the PSF does not smoothly vary across the image. The blur kernel may vary smoothly within the segments of the image, but it may change abruptly between segments. This chapter presents a new PSF estimation strategy with assumptions that the blur kernel is uniform within a local region and that the local region affected by any blur kernel is of sufficient size and with rich texture as well. The proposed framework contains

three main steps. The first step is to obtain coarse estimates of the blur kernels. The input image is splitted into patches of equal size. For each patch, the blur kernel is estimated using a fast space-invariant PSF estimation technique by Cho [37]. For patches with uniform blur and sufficient spatial variance, we expect to have good PSF estimates. For patches that include regions which have different blur kernels or for patches that do not have spatial variance, the PSF estimates would not be accurate. The second step is to refine the blur kernels through clustering. As we assume that there are several distinct blur kernels, we expect the coarse PSF estimates obtained in the first step cluster around true blur kernels. I use the K-means clustering algorithm and obtain the cluster centroids. The blur kernels that are close to a cluster centroid are likely to be good estimates; the kernels that are not close to a cluster centroid are likely to correspond to patches with nonuniform blur or insufficient spatial variance. The patches corresponding to kernels that are close to a cluster centroid are merged to form larger regions. The merged regions are expected to have a uniform blur kernel. Once PSFs are estimated using merged regions separately, space-varying image deblurring is performed through image fusion in the third step, which produces a high-quality image even for regions where it is not clear what the correct PSF is at first.

This chapter is organized as follows: Section 4.2 will propose the space-varying image deblurring framework. In Section 4.3, experiments will be conducted on some images degraded by space-varying blur. Comparisons with some state-of-the-art motion deblurring algorithms will also be provided. In addition, I will discuss some limitations of our proposed method in this section as well. In Section 4.4, I will conclude our work and discuss my future work to improve the proposed space-varying image deblurring algorithm.

4.2 The proposed framework

The proposed framework includes three main steps: (1) Coarse PSF estimation, (2) PSF refinement through kernel clustering, and (3) Space-varying deblurring through image fusion.

The blurred image degraded by space-varying blur kernel is denoted as g. The first step spits the whole image g into N patches of equal size g_i and then estimates the PSF for each patch separately using a modified version of Cho's method [37], resulting in N blur PSFs h_i. The second step is to refine PSF estimation. The N blur kernels h_i are clustered into K classes through K-means clustering. Here K denotes the number of distinct blur kernels across the whole image. The K blur kernels obtained from K-means clustering are refined for the second time by first detecting reliable regions which are composed of patches closer to centroids for uniform blur estimation and estimating blur kernels again with larger region using Cho's method [37], resulting in K refined blur kernels h_i^{true}. Then in the last step, the final image deblurring through image fusion is performed which produces good results even for regions where it is not clear what the correct PSF is at first. The three steps will be detailed as follows.

4.2.1 Coarse point spread function (PSF) estimation

With the assumption that the blur kernel is uniform within local region, I split the input image into patches and estimate each patch separately. However, the difficulty is how to decide the size of each patch. Small patches better approximate local space-invariant blur kernel. However, on the other hand, small patches are not large enough to contain information to estimate PSFs. As Sroubek [86] pointed out, the necessary minimum patch size decreases with the decreasing PSF size, but prior information about the size and nature of PSF should be known beforehand. In this chapter, I use overlapping window to split the whole image g into N patches of equal size g_i. The reason why I choose the overlapping window is that more blur kernel samples can be obtained to make K-meaning clustering more accurate with enough data samples. However, the overlapping window should also be carefully designed. If the window is shifted slightly, then more patches can be obtained, however, more patches mean more computational overhead for PSF estimation. On the other hand, if the window is shifted largely, the splitted patches are more likely to contain non-uniform blur. In my

experiment, the window is shifted half of the size of patch horizontally and vertically. Also, the size of patch is determined by the nature of the blur which most degrades the local region among all distinct blur kernels. For example, if an image is degraded by space-varying blur which includes large motion blur and small motion blur, then the size of each patch is decided by the nature of large motion blur. This strategy will reduce the PSF estimation error if we choose the other way around due to small size of patch. For each patch, I make no assumption of any blur type and utilize an improved version of Cho's method [37] to separately estimate coarse blur PSF h_i of each patch g_i. The blur process for each patch g under uniform blur kernel h is modeled as follows:

$$g = f \otimes h + n, \tag{4.1}$$

where f is the latent image, $\otimes$ operator denotes convolution and n is the additional noise. For the convenience of mathematical description, I omit subscript i for each patch. The coarse PSF estimation adopts coarse-to-fine image pyramid strategy to progressively refine the blur kernel. In each pyramid, the estimation method iterates three steps: strong edge prediction, fast kernel estimation and coarse image deconvolution. The above iteration process stops at the highest level of image pyramid which has the equal size of the input blurred image. Finally, blur kernel sparsity is enforced through hard-thresholding, which sets the kernel values that are less than the 10 percent of the maximum value in the kernel to be 0.

4.2.2 Point spread function (PSF) refinement through kernel clustering

The first step separately estimates blur kernels h_i for each of N splitted patches g_i. As I assume that there are several distinct blur kernels, the coarse PSF estimates obtained in the first step are expected to cluster around true blur kernels. The number of true blur kernels across the entire image may be determined beforehand by assuming the known number of distinct blur kernels, although the regions affected by these blur kernels are not known beforehand. This assumption is reasonable because we may intuitively know how many different

blur kernels degrade an image by our naked eyes. Here, I set the number of true blur kernels affecting the entire image to be K. Then I use the K-means clustering algorithm [87] to partition N blur kernels into K classes and obtain the K cluster centroids $h_k^{centroid}$. The blur kernels that are close to a cluster centroid $h_k^{centroid}$ are likely to be good estimates; the kernels that are not close to a cluster centroid are likely to correspond to patches with nonuniform blur or insufficient spatial variance. The patches corresponding to kernels that are close to a cluster centroid $h_k^{centroid}$ are merged to form larger regions $g_k^{centroid}$. The larger region $g_k^{centroid}$ is also considered to be reliable region of the true uniform blur kernel. The cluster centroid $h_k^{centroid}$ may be refined by estimation from the larger region $g_k^{centroid}$, resulting in true blur kernel h_k^{true}. Again Cho's method [37] is used for estimation. We expect more accurate blur kernel than the cluster centroid because a larger region is used which contains more texture information. Also, actually the cluster centroid is the closest data to all samples assigned to that blur kernel class. However, these samples may also include samples that contain non-uniform blur in that region and those that have little spatial invariance. The estimated cluster centroid may be degraded by these samples as well. Thus, reliable regions need to be used to refine the coarse blur kernel from the first step.

4.2.3 Space-varying deblurring through image fusion

The final step is to perform space-varying image deblurring. The process so far estimates the blur kernels h_i^{true} but we do not have a blur kernel assigned to each pixel. Only the reliable regions with uniform blur have known kernels. However, for regions where no blur kernel is decided, its true blur kernel needs to be determined and then the entire image needs to be restored. To restore the entire image, the entire image is deblurred for each kernel and then the deblurred images are fused into a single all-deblurred image. I start by deblurring the entire image g with each of the K refined kernels h_i^{true} obtained in the last step, providing K deblurred images $\hat{f}_i$. For the image deconvolution process, to achieve robustness to outliers as well as sharp results, the TV-L1 based method proposed by Wang [55] is implemented,

which minimizes the following energy function

$$E(\hat{f}_i) = \|g - h_i^{true} \otimes \hat{f}_i\|_1 + \lambda \|\nabla \hat{f}_i\|_2, \tag{4.2}$$

where h_i^{true} is the estimated blur kernel, g is the original blurred image, $\hat{f}_i$ is the latent image to be estimated, and λ is the regularization parameter. Clearly, it brings non-linearity and non-differentiability to both data and regularization terms, resulting in computational difficulty to solve this problem. To overcome the computational difficulties, Wang [55] proposed an alternating minimization method based on half quadratic splitting to solve the TV-L1 optimization problem. I follow the exact algorithm [55] to achieve motion deblurring. In my implementation, the regularization parameter λ is set to be 10.

Next, I will use these deblurred images to produce a high quality image. For each blur kernel h_i^{true}, the robust reconstruction error is computed. The robust reconstruction error includes two terms: data fidelity term $e_{i,data} = g - h_i^{true} \otimes \hat{f}_i$ and regularization term $e_{i,reg} = \nabla \hat{f}_i$. A deblurred image $\hat{f}_i$ will usually provide a smooth plausible reconstruction for parts of the image where h_i^{true} is the true blur kernel. The reconstruction in other areas, whose blur kernel differs from h_i^{true}, will contain serious ringing artifacts since those areas cannot be plausibly explained by the kernel h_i^{true}. These artifacts ensure that the robust reconstruction error for such areas will be high. I compute a local approximation for the energy around the pixel position $\mathbf{x}$, by summing up the robust reconstruction error over a small local window $W_{\mathbf{x}}$ as follows:

$$\hat{E}_i(\mathbf{x}) = \sum_{\mathbf{y} \in W_{\mathbf{x}}} e_{i,data}^2(\mathbf{y}) + \lambda |e_{i,reg}(\mathbf{y})|. \tag{4.3}$$

In my experiment, the size of small local window $W_{\mathbf{x}}$ is equal to that of blur kernel PSF. The local energy estimate is then used to select the blur kernel in the pixel $\mathbf{x}$. The blur kernel with the smallest robust reconstruction error is chosen as the kernel in the pixel $\mathbf{x}$ as follows:

$$kernel(\mathbf{x}) = \arg\min_i \hat{E}_i(\mathbf{x}). \tag{4.4}$$

High quality image is produced through image fusion by picking each pixel independently from one of the K deblurred images $f^{restored}(\mathbf{x}) = \hat{f}_{kernel(\mathbf{x})}(\mathbf{x})$. Here, $f^{restored}(\mathbf{x})$ denotes the final deblurred image in the $\mathbf{x}$ pixel position. The proposed framework is illustrated in Figure 4.1.

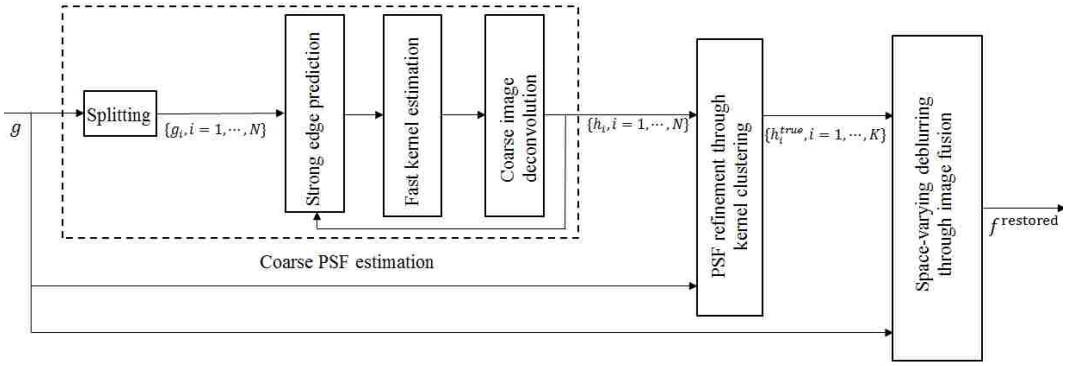


FIGURE 4.1. The framework for proposed space-varying deblurring method

4.3 Experiments and discussions

In this section, some experiments are conducted to illustrate our proposed space-varying image deblurring method. In the experiment, in order to obtain a blurred image degraded by space-varying blur, a blurred image is captured by a Canon camera(in Figure 4.2(a)) in which both motion blur and defocus blur, that is, two distinct blur kernels exist in the scene. The motion blur is caused by moving object (book in Figure 4.2(a)) when I take a photo of the scene during exposure. On the other hand, the defocus blur is caused by camera's limited depth of field (DOF) when the camera's focus area is set on the moving object, leaving other depth-variant objects in the scene to be out of focus. To perform the coarse PSF estimation as presented in the last section, I first divide the whole image into 11×11 overlapping patches of equal size. The size of the whole blurred image is 1872×1248, whereas the size of each patch is 312×208. That means, I move the overlapping window across the entire image by 156 pixels in horizontal direction and 104 pixels in vertical direction to get these overlapped patches.

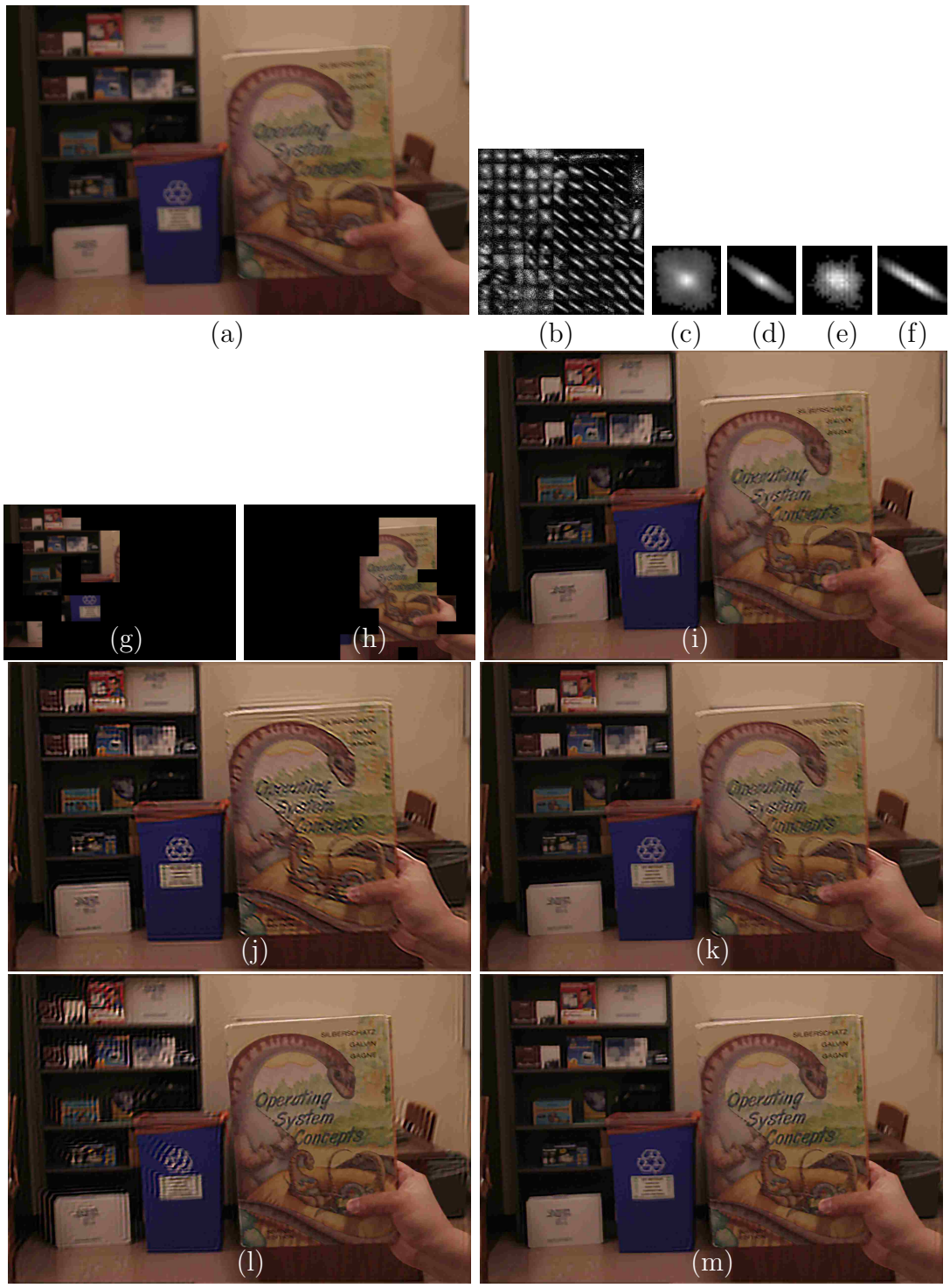


FIGURE 4.2. Experimental results of the proposed method. (a) Original space-varying blurred image degraded by motion blur and defocus blur; (b) 11×11 estimated blur kernels for each split by fast coarse PSF estimation; (c) and (d) Coarse defocus blur and motion blur by K-means clustering method; (e) and (f) Refined defocus blur and motion blur; (g) and (h) Detected reliable region of uniform defocus blur and motion blur; (i) and (j) Deblurred images using coarse defocus blur and motion blur((c) and (d)), respectively; (k) and (l) Deblurred images using refined defocus blur and motion blur((e) and (f)), respectively; (m) Final restored image by merging (k) and (l) based on proposed local PSF selection method.

For each patch, I estimate the coarse PSF using fast blind kernel estimation method by Cho [37]. The 121 blur kernels are shown in Figure 4.2(b). Then K-means clustering is applied to 121 blur kernels, where the number of blur kernel classes K is set to be two since we have two distinct blur kernels across the entire image. The two cluster centroids which are also coarse defocus blur and motion blur are given in Figure 4.2(c) and 4.2(d). The merged regions affected by kernels that are also close to the centroids in Figure 4.2(c) and 4.2(d) are shown in Figure 4.2(g) and 4.2(h), respectively. These two merged regions are also considered to be affected by two uniform blur kernels, respectively. The uniform blur kernels, that is, defocus blur and motion blur, are estimated for a second time from these two merged regions. As these regions are larger than the original patches, we expect more accurate kernel estimates. The refined blur kernels are shown in Figures 4.2(e) and 4.2(f). Figure 4.2(i) is the deblurred image when the kernel in Figure 4.2(c) is used, and Figure 4.2(j) is the deblurred image when the kernel in Figure 4.2(d) is used. Figure 4.2(k) is the deblurred image when the kernel in Figure 4.2(e) is used, and Figure 4.2(l) is the deblurred image when the kernel in Figure 4.2(f) is used. By comparison of deblurred images before and after PSF refinement, we can see that the PSF refinement can better approximate the true kernels. In addition, as also seen in these figures (in Figures 4.2(k) and Figures 4.2(l)), there are distinct regions a specific kernel worked better, and I fuse these images to form a all-deblurred image according to the proposed image fusion method in the last section. The fused image is shown in Figure 4.2(m). Figure 4.3 includes zoomed-in regions from the experiment. Figure 4.3(a) is from the original input image; clearly, both regions are blurry. Figure 4.3(b) is from the deblurred image given in Figure 4.2(i), whereas Figure 4.3(c) is from the deblurred image given in Figure 4.2(j). From these zoom-in regions, it shows that, although the cluster centroid can provide a rough estimate for blur kernel, it is not accurate enough to be used as true blur kernel to restore the blurred image. This necessitates the step of the PSF refinement through kernel clustering. Figure 4.3(d) is from the deblurred image

given in Figure 4.2(k); while one region is deblurred well, the other region is not. Figure 4.3(e) is from the deblurred image given in Figure 4.2(l). The deblurred regions are opposite to those shown in Figure 4.3(d). Figure 4.3(f) is from the final deblurred image given in Figure 4.2(m), which merges two deblurred images in Figures 4.2(k) and 4.2(l). As seen, both regions are well deblurred, showing that the proposed image fusion algorithm performs well in this example.

FIGURE 4.3. Zoomed-in regions. (a) Input image, same as in Figure 4.2(a); (b) Deblurred image, given in Figure 4.2(i); (c) Deblurred image, given in Figure 4.2(j); (d) Deblurred image, given in Figure 4.2(k); (e) Deblurred image, given in Figure 4.2(l); (f) Final deblurred image given in Figure 4.2(m).

In the second blurred image (shown in Figure 4.4(b)) degraded by space-varying blur, the regions containing the car are affected by defocus blur, whereas the region containing the book is degraded by motion blur. In addition, I compare our method with some state-of-the-art blind image deblurring algorithms (in Figures 4.4 and 4.5). I deblur the blurred image using Cho's method [37], Xu's method [45] and Shan's method [34]. The comparison shows that my proposed method can better restore the original image with fewer unwanted artifacts from one single image degraded by space-varying blur.

In my proposed method for space-varying image deblurring problem, I assume no space-varying blur type which works well in a variety of general cases. The underlying assumption is that each distinct blur kernel is uniform in the local region which is also of sufficient size and with rich texture to get accurate PSF estimation. In addition, to get more samples for K-mean clustering in the PSF refinement step, I use overlapping window to split the entire image into patches. However, I choose the size of patch and shift by trial and error. I did not study how to robustly decide these two important parameters in my proposed method. Also, when I merge patches to obtain reliable regions for PSF refinement, I choose the patches closer to centroid as good candidates. However, I did not give a metric to decide the threshold for good candidates. In addition, I consider neither boundary information nor object segmentation when segmenting different regions affected by distinct blur kernels, resulting in rough and inaccurate segmentation around the boundaries where blur kernel changes abruptly. However, our result may be improved by introducing human assistance by inferring different regions affected by distinct blur kernels by hand in my future work.

4.4 Conclusions and future work

In this chapter, I investigate image deblurring problem from one single image degraded by space-varying blur across the whole image. The proposed method makes neither assumption of the blur type nor how blur kernel changes across the whole image. However, the proposed

method assumes that the blur kernel is uniform within local region of sufficient size and rich texture.

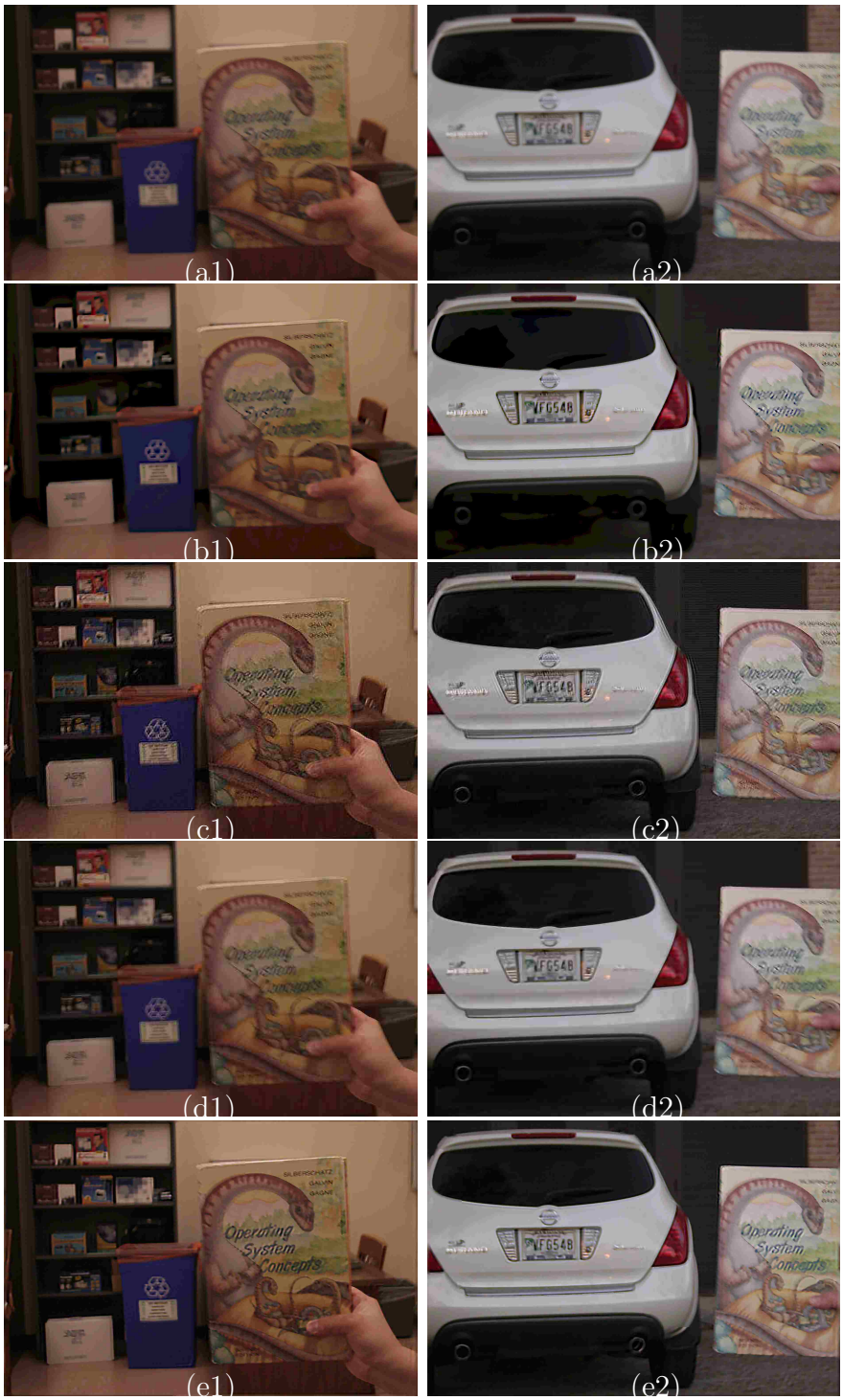


FIGURE 4.4. Comparison of state-of-the-art blind image deblurring algorithms. (a1) and (a2) Two original images; (b1) and (b2) Cho's method [37]; (c1) and (c2) Shan's method [34]; (d1) and (d2) Xu's [45] method; (e1) and (e2) My proposed method

FIGURE 4.5. Comparison of state-of-the-art blind image deblurring algorithms (Zoomed-in regions). (a1) and (a2) Two original space-varying blurred images; (b1) and (b2) Deblurred images using Cho's method [37]; (c1) and (c2) Deblurred images using Shan's method [34]; (d1) and (d2) Deblurred images using Xu's [45] method; (e1) and (e2) Deblurred images using my method;

The method first progressively estimates distinct kernels as well as regions affected by blur kernels. Once accurate estimation of blur kernel is obtained, space-varying image deblurring is performed in the last step, which produces good results even for regions where it is not clear what the correct PSF is at first. The last image fusion step utilizes the reconstruction error as metric for good PSF candidate to get restored image. The experimental results and comparison with other state-of-the-art methods illustrate the effectiveness of my proposed method.

However, to get more accurate PSF and region segmentation of distinct blur kernels in cases that the underlying assumption does not hold well, more future work needs to be done to improve the proposed method. As future work, I will take advantage of object structures to guide the blur classification, and object information may be used for blur segmentation. Another simple way to improve the proposed method is to introduce human assistance for blur PSF estimation and blur kernel clustering by manually and roughly segmenting the regions affected by distinct blur kernels.

5 Extending depth of field and dynamic range from differently focused and exposed images

5.1 Introduction

Although many sophisticated algorithms have been proposed for either focus stacking or high dynamic range (HDR) image creation, extending both depth of field (DOF) and dynamic range from differently exposed and focused images has seldom been investigated. Qian and Gunturk [25] presented a joint HDR merging and focus stacking algorithm to extend depth of field and dynamic range. The method introduced the weighting map which includes both exposure-based weighting map and local sharpness-based map so that more weights will be given to in-focus and suitably exposed regions when estimating HDR radiance map. In addition, the proposed joint HDR creation and focus stacking has also been implemented on Android phone. However, it is nontrivial to determine optimal values since in some cases the weighting map based on local sharpness or exposure information is decided by the scene case-to-case. This method has not yet been robust enough to adaptively detect in-focus and suitably exposed regions. Therefore we need to adjust the balancing factors by trial and error. In this chapter, instead, I will propose another method which produces an image with both extended dynamic range and DOF from differently focused and exposed Images. The proposed method mainly includes three steps: (1) spatial registration, (2) focus stacking under exposure diversity, and (3) HDR image creation. Both focus stacking and HDR imaging becomes challenging when the camera or the scene is not static. It is indeed rarely the case

that the camera and the scene are static. In focus stacking, even when the camera is fixed, the lens compound may cause geometric distortion as the focus depth is changed. In general, we need to estimate dense motion field for both focus stacking and HDR imaging. First, as I have already mentioned, in order to get the same scene for extending DOF and dynamic range, accurate image registration is essential. Here I employ the spatial registration approach [88] that I have recently investigated. The histogram-based intensity mapping function (IMF) estimation [89] is first used for photometric registration and then an advanced optical flow method [61] is used for geometric registration. This optical flow method is based on a robust data term and discontinuity-preserving total-variation (TV) regularization term and results in accurate dense flow fields. After the images are registered, focus stacking under exposure diversity is first performed. Although many focus stacking algorithms have been proposed to extend DOF from multiple images with different in-focus regions, these algorithms have two main requirements for input image stacks. First, in-focus region for all image stacks is unique. This requirement ensures that in the image fusion step, each pixel in the reconstructed image comes from unique slice of image stacks based on the maximum value selection rule. Second, the illumination should be approximately constant across all image stacks. Unfortunately, these two requirements do not hold in this chapter as in-focus regions may exist in more than one image stack, and image stacks may have different exposure information as well. Then, with the application of my proposed focus stacking from differently exposed images, an all-in-focus image is obtained from well registered images with different in-focus regions and exposure information. Instead of using the typical existing maximum selection rule, the weighting map rule is used to indicate the degree of focus from image stacks. The weighting map rule is applied here because of the fact that in-focus regions may exist in more than one image with different exposure information. In this case, simple maximum selection rule will lose useful sharp information from images in-focus but not selected. The weighting maps are initially estimated by classical local sharpness method and refined by a reliability mask

which will be discussed in details later. The all-in-focus images are created by the proposed focus stacking method, which will then be used to estimate the HDR radiance map; finally, a local tone mapping operator [80] is used to display the radiance map on low dynamic range devices. The resulting image has both extended DOF and dynamic range.

This chapter is organized as follows: Section 5.2 will introduce the proposed method in which spatial registration, focus stacking and HDR imaging algorithms are described in details. In Section 5.3, I will conduct two experiments to demonstrate my proposed method. Section 5.4 will conclude this chapter.

5.2 Focus stacking under exposure diversity and high dynamic range (HDR) creation

In this section my approach to achieving focus stacking is explained when the input images do not necessarily have the same exposure settings. I first present the method assuming the input images are grayscale, with the purpose of notational simplicity; and at the end I explain how to treat color images. The flow of my method is illustrated in Figure 5.1. The input images $\mathbf{f}_1, \cdots, \mathbf{f}_N$ are first spatially registered using the reference image $\mathbf{f}_r$, which is chosen among the input images. Spatial registration is to handle cases where the scene and/or the camera are not static. The registered images $\widetilde{\mathbf{f}}_1, \cdots, \widetilde{\mathbf{f}}_N$ are then photometrically mapped to the reference image $\widetilde{\mathbf{f}}_k$, which is chosen among the registered images. In this step, I also calculate weight maps for each image. The weight maps take two things into account: local sharpness and registration errors. Local sharpness is an indicator of focus; if a pixel is in an in-focus area, its local sharpness is large. Registration error is an indicator about the accuracies of spatial and photometric registration steps. The weight of a pixel is linearly proportional with the local sharpness and inversely proportional with the registration error. The photometrically mapped images $p_{ik}(\widetilde{\mathbf{f}}_i), \cdots, p_{ik}(\widetilde{\mathbf{f}}_N)$ are merged by pixel-by-pixel weighted sum to produce the focus stacked image $\mathbf{f}_k^{focus}$. Now these steps are explained in details.

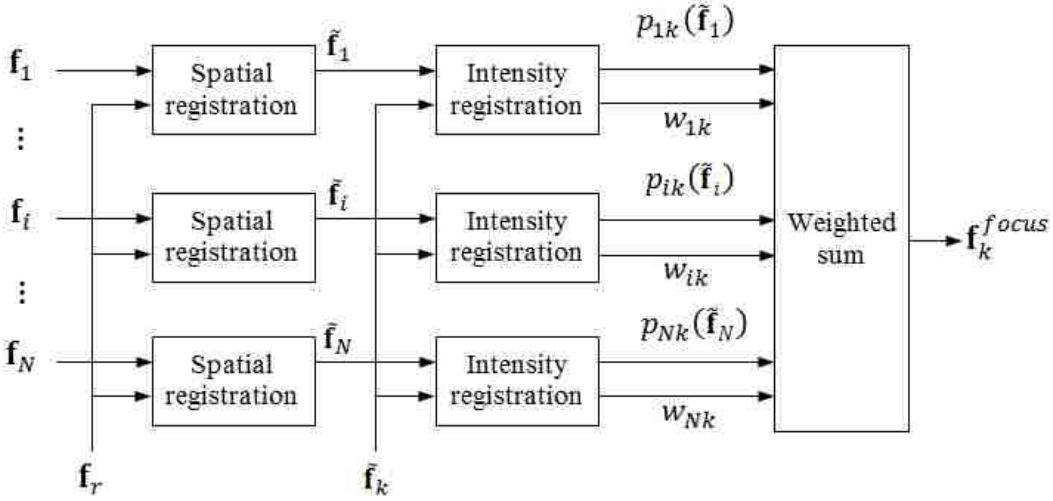


FIGURE 5.1. Focus stacking from differently exposed images

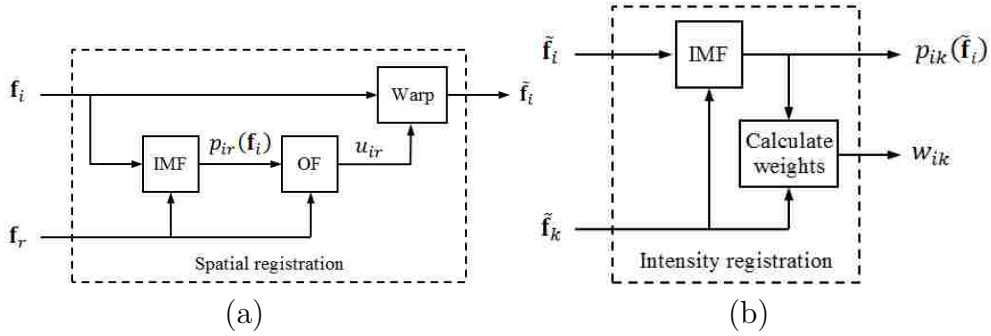
5.2.1 Spatial registration

Suppose we have N grayscale images $\mathbf{f}_i,\ i = 1, \cdots, N$ with different in-focus regions and exposure settings. My first task is to estimate the motion field and register the images. I start with choosing a reference image $\mathbf{f}_r$ among the input images. The optical flow equation between an input image $\mathbf{f}_i$ and the reference image $\mathbf{f}_r$ should reflect the intensity mapping to take care of different exposure settings and can be formulated as

$$p_{ir}(\mathbf{f}_i(\mathbf{x} + u_{ir}(\mathbf{x}))) = \mathbf{f}_r(\mathbf{x}), \tag{5.1}$$

where $\mathbf{x}$ is a pixel position, $u_{ir}(\mathbf{x})$ is the motion vector at pixel $\mathbf{x}$ from input image $\mathbf{f}_i$ to the reference image $\mathbf{f}_r$, and $p_{ir}(\cdot)$ is the intensity mapping function (IMF) from $\mathbf{f}_i$ to $\mathbf{f}_r$. (Note that when two images have the same exposure settings, $p_{ir}(\cdot)$ is identity function and equation (5.1) reduces to the standard optical flow equation.) Once the IMF is applied to an input image, the photometrically mapped image $p_{ir}(\mathbf{f}_i)$ and the reference image $\mathbf{f}_r$ satisfy the constant brightness assumption; and therefore, standard optical flow estimation methods can be utilized to estimate the motion field u_{ir}. In my implementation, I estimated the IMF using the histogram-based method of Mitsunaga and Nayar [11] and the motion field using the method of Liu [61]. Using the estimated motion field, I warp the input image $\mathbf{f}_i$ onto the

reference image $\mathbf{f}_r$. The warped images $\widetilde{\mathbf{f}}_i,\ i = 1, \cdots, N$ are now expected to be spatially registered. The flowchart of the spatial registration process is given in Figure 5.2(a).



(a) (b)

FIGURE 5.2. (a) Spatial registration of differently exposed images. An input image $\mathbf{f}_i$ is first photometrically registered to the reference image $\mathbf{f}_r$ by applying the intensity mapping function (IMF). The motion field u_{ir} is then estimated by applying an optical flow estimation method. The input image $\mathbf{f}_i$ is finally warped to achieve spatial registration. (b) Intensity mapping and weight calculation.

5.2.2 Intensity registration and focus stacking

The images $\widetilde{\mathbf{f}}_i,\ i = 1, \cdots, N$ are spatially registered but with possibly different exposure settings and focus areas. The next step is to produce an image with extended depth of field. Standard focus stacking methods cannot be applied to $\widetilde{\mathbf{f}}_i,\ i = 1, \cdots, N$ because they might have different exposure settings. One may suggest to apply a standard focus stacking to photometrically mapped images $p_{ir}(\widetilde{\mathbf{f}}_i)$; however, this approach would not work in general either because saturated regions in a long-exposure image cannot photometrically mapped to corresponding unsaturated regions in shorter-exposure images. To handle complications due to different exposure settings, I propose the following approach.

I choose a reference image $\widetilde{\mathbf{f}}_k$ among the spatially registered images $\widetilde{\mathbf{f}}_i,\ i = 1, \cdots, N$. I estimate the IMF $p_{ik}(\cdot)$ and obtain photometrically registered images $p_{ik}(\widetilde{\mathbf{f}}_i)$. (Note that $p_{kk}(\cdot)$ is identity function.) As mentioned earlier, the focus stacked image $\mathbf{f}_k^{focus}$ is obtained as a weighted sum of $p_{ik}(\widetilde{\mathbf{f}}_i), i = 1, \cdots, N$. The weights reflect two things at each pixel: (1) local sharpness, which is expected to be correlated with focus, and (2) spatial and photometric

registration errors. We would like to have a large weight for a pixel that is in an in-focus area and has low spatial/photometric registration error.

The use of local sharpness as an indicator of focus is a commonly used idea in focus stacking. The local sharpness $\widetilde{s}_{ik}(\mathbf{x})$ at pixel $\mathbf{x}$ of image $p_{ik}(\widetilde{\mathbf{f}}_i)$ is defined as

$$\widetilde{s}_{ik}(\mathbf{x}) = \sum_{\mathbf{y} \in N_h(\mathbf{x})} \|\nabla p_{ik}(\widetilde{\mathbf{f}}_i(\mathbf{y}))\|, \tag{5.2}$$

where $\nabla p_{ik}(\widetilde{\mathbf{f}}_i)(\mathbf{y})$ is the gradient vector at pixel $\mathbf{y}$ obtained by applying a Sobel filter, $\|\cdot\|$ denotes the gradient magnitude, and $N_h(\mathbf{x})$ is a $h \times h$ window around pixel $\mathbf{x}$.

Now we have spatially and photometrically registered images $p_{ik}(\widetilde{\mathbf{f}}_i)$ and the corresponding weight maps $\widetilde{s}_{ik}$ indicating in-focus regions for each image. These photometrically mapped images are expected to have the same illumination. However, the zero-slope or near zero-slope regions in IMF will cause saturation problems in the photometrically mapped images [90]. This is because in saturated regions in the target image $\widetilde{\mathbf{f}}_i$, there is no corresponding information in the reference image $\widetilde{\mathbf{f}}_k$. Therefore, one-to-one photometric mapping is not possible. The saturated regions are not reliable when I merge the photometrically mapped images to create all-in-focus image. Also, the absolute difference of intensity values between $p_{ik}(\widetilde{\mathbf{f}}_i)$ and $\widetilde{\mathbf{f}}_k$ shows that outliers caused by photometric mapping error have larger values. Hence it is necessary to reject these outliers when merging these pixels to create fused images, which could be achieved through comparing with a threshold. However, in-focus and out-of-focus pixels also have a relatively large absolute difference of intensity values. The threshold value should be carefully chosen so that it can reject outliers as well as preserve in-focus pixels. Thus, when I merge all photometrically mapped images to create all-in-focus image, a reliability mask $M_{ik}(\mathbf{x})$ is introduced to rule out saturated regions for photometrically mapped image $p_{ik}(\widetilde{\mathbf{f}}_i)$ defined as follows:

$$M_{ik}(\mathbf{x}) = H(T - |p_{ik}(\widetilde{\mathbf{f}}_i(\mathbf{x})) - \widetilde{\mathbf{f}}_k(\mathbf{x})|), \tag{5.3}$$

where $H(\cdot)$ is the Heaviside step function, outputting zeros for negative values and ones otherwise. (Note that $M_{kk}(\mathbf{x}) = 1$.) T is intensity breakpoint for pixel reliability determined by trial and error in our case. The weight map for input image $\{\widetilde{\mathbf{f}}_i\}$ is defined as

$$\widetilde{w}_{ik}(\mathbf{x}) = \widetilde{s}_{ik}(\mathbf{x}) M_{ik}(\mathbf{x}). \tag{5.4}$$

Before I do the image fusion, I need to normalize the calculated weight maps:

$$w_{ik}(\mathbf{x}) = \frac{\widetilde{w}_{ik}(\mathbf{x})}{\sum_{i=1}^{N} \widetilde{w}_{ik}(\mathbf{x})}, \tag{5.5}$$

where $i = 1, \cdots, N$. When all weights are zero, then division by zero occurs in the above normalization. We have two options, the first option is to assign $1/N$ to each weight. The second option is to assign 1 to the weight associated with the reference image, and 0 to all other target images. In this chapter, I choose the second option because I trust the reference image more than other target images for possible saturation problems in the photometrically mapped target images when weights are all zero. I achieve this by adding a small scalar number to the weight associated with the reference image before the normalization:

$$\widetilde{w}_{kk}(\mathbf{x}) = \widetilde{s}_{kk}(\mathbf{x}) + \epsilon, \tag{5.6}$$

where ϵ is a small scalar number, which is set as 0.1 in our paper. Lastly, the image fusion is performed to get the all-in-focus image $\mathbf{f}_k^{focus}$ by merging all photometrically mapped images:

$$\mathbf{f}_k^{focus}(\mathbf{x}) = \sum_{i=1}^{N} w_{ik}(\mathbf{x}) p_{ik}(\widetilde{\mathbf{f}}_i(\mathbf{x})), \tag{5.7}$$

The flowchart for intensity registration and weight calculation is given in Figure 5.2(b).

5.2.3 Extension to color images

So far, I discussed that method for grayscale images. It may be extended to the color images as follows. For spatial registration, the green channel of input images are used to estimate the intensity mapping function and motion vectors between the input images and reference

image. Using the motion vectors estimated from the green channel, all three channels are warped onto the color reference image. For photometric registration and focus stacking, each color channel is processed separately. That is, IMF and weight maps are estimated for each channel, and the weighted sum process is performed to obtain focus stacked version of each channel.

5.2.4 High dynamic range (HDR) radiance estimation and tone mapping

$\mathbf{f}_k^{focus}$ is the all-in-focus image where the kth image and therefore its photometric settings are used as the reference. If we have N images, then the focus stacking process may be repeated for each of those N images. The resulting all-in-focus images can then be processed to obtain an HDR image using a standard HDR creation algorithm. In this chapter, I used Debevec's algorithm [9] to estimate the HDR radiance map. After the HDR radiance map is obtained, tone mapping should be operated for low dynamic range displays. In this chapter, I utilize Fattal's local tone mapping method [80] to display the images. Although a global tone mapping operator such as Reinhard's method [79] is simple and fast, local tone mapping operator can greatly reduce the contrast while preserving details well but at the cost of computational complexity and annoying parameter tuning for best results.

5.3 Experimental results

I provide experimental results with real data to illustrate my proposed method for extending DOF and dynamic range from differently exposed and focused images. In my first experiment, I captured four images with a hand-held DSLR camera. Each image has a different focus area and exposure time combination. These images are given in Figure 5.3. Figure 5.3(a) has exposure time of 1/200 seconds, and is near-focused; Figure 5.3(b) has exposure time of 1/200 seconds, and is far-focused; Figure 5.3(c) has exposure time of 1/1600 seconds, and is far-focused; and finally, Figure 5.3(d) has exposure time of 1/1600 seconds, and is near-focused. (Note that the details of dark regions of both original images and created all-

in-focus images may be lost in the printed hardcopy version, However, the details are clear in the electronic version for these images.)

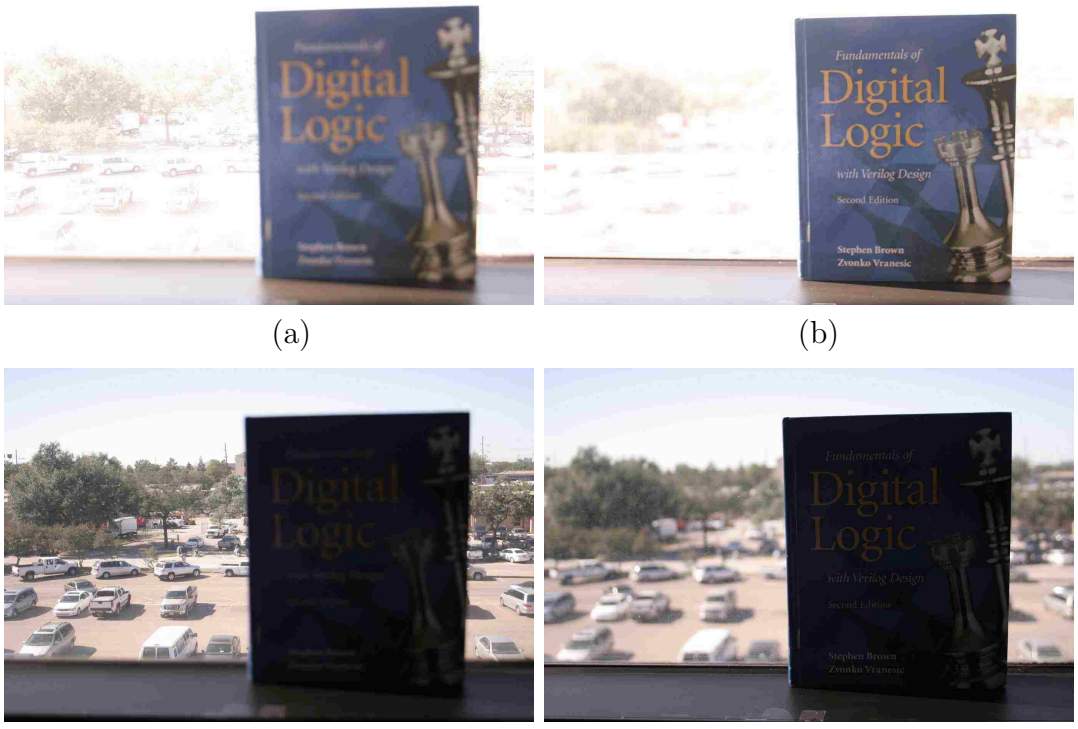


FIGURE 5.3. Original four images with different focus regions and exposure levels. (a) Reference image for image registration; (b) Target image 1 for image registration; (c) Target image 2 for image registration; (d) Target image 3 for image registration; ((a) and (b) have different focus regions with the same exposure time of 1/200 second; (c) and (d) have different focus regions with the same exposure time of 1/1600 second.)

Figure 5.4 shows the spatial registration process. The first row shows the luminance channels of the input images given in Figure 5.3. The first image $\mathbf{f}_1$ is set as the reference image. The second row shows the IMFs $p_{ir}(\cdot)$. The third row shows the photometrically registered images $p_{ir}(\mathbf{f}_i)$. And finally, the fourth row shows the estimated motion fields u_{ir} obtained from the photometrically registered images $p_{ir}(\mathbf{f}_i)$.

Spatial registration is done by warping all three channels using the estimated motion fields. The spatially registered images are shown in Figure 5.5. (Note that the first image is the reference image, therefore, it is not warped.)

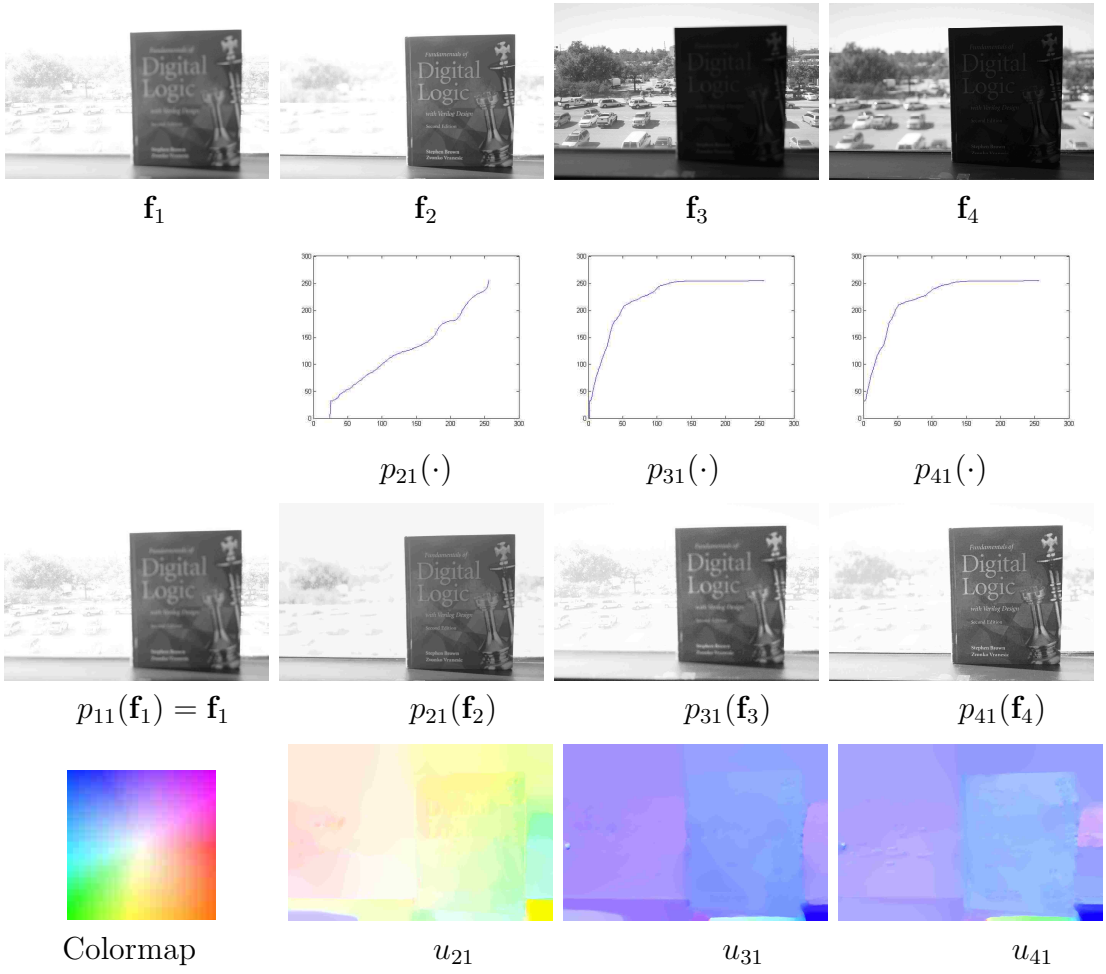
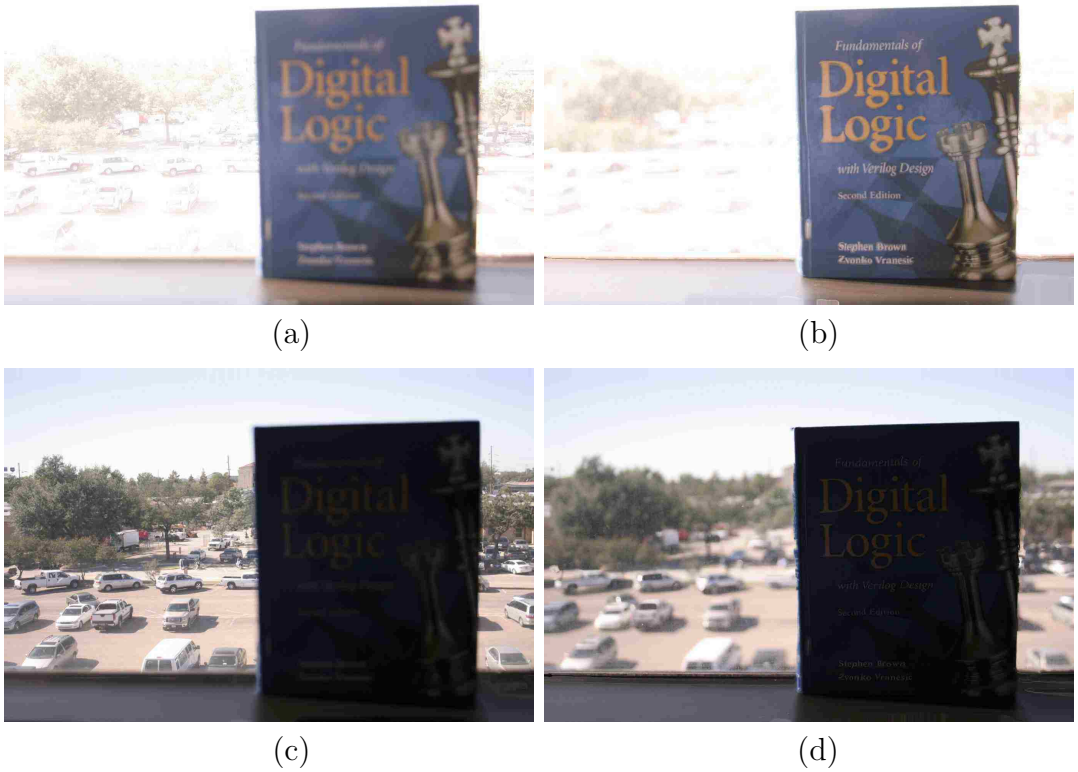


FIGURE 5.4. Spatial registration. First row: grayscale versions of images in Figure 5.3; Second row: Estimated intensity mapping functions (IMFs) from target images in Figures 5.3(b-d) to reference image in Figure 5.3(a); Third row: Photometrically mapped images; Fourth row: Color coding of motion vectors and estimated motion fields between the target images and reference image.

The accuracy of the spatial registration may not be obvious from Figures 5.4 and 5.5. Therefore, I include Figure 5.6, which shows the absolute differences between the reference image and the input images. Figures 5.6(a1)-5.6(a3) show the differences between the input images without any spatial registration. Figures 5.6(b1)-5.6(b3) show the differences after photometric registration (IMF). The residuals are reduced compared to Figures 5.6(a1)-5.6(a3); however, it is obvious that there is some movement between the images. Figures 5.6(c1)-5.6(c3) show the differences after both IMF and optical flow method for spatial registration. The residuals are reduced significantly, demonstrating the effectiveness of the



(a) (b)

(c) (d)

FIGURE 5.5. Spatially registered images. (a) Original reference image in Figure 5.3(a); (b-d) Geometrically registered images from target images in Figures 5.3(b)-5.3(d) to Figure 5.3(a) based on the estimated motion vectors in the fourth row of Figure 5.4.

registration process. Table 5.1 also shows mean absolute differences (MADs) between three input images and reference image before registration, after IMF and after both IMF and optical flow, respectively. The MAD is greatly decreased after the introduced spatial registration. This completes the registration step.

Figures 5.7, 5.8, 5.9 and 5.10 show results of focus stacking under exposure diversity. I use the four registered images in Figure 5.5 as input images for focus stacking. As I mentioned

TABLE 5.1. Mean absolute difference of intensity values between three target images and reference image for spatial registration (The intensity range is [0, 255].)

Image pair	**No registration**	**IMF only**	**IMF+OF**
Target image1	9.4	12.8	6.9
Target image2	82.6	18.0	3.0
Target image3	80.6	17.3	5.1

previously, the four images in Figure 5.5 have two exposure levels (1/200 second and 1/1600 second). In addition, within each exposure level, in-focus regions exist in different images.

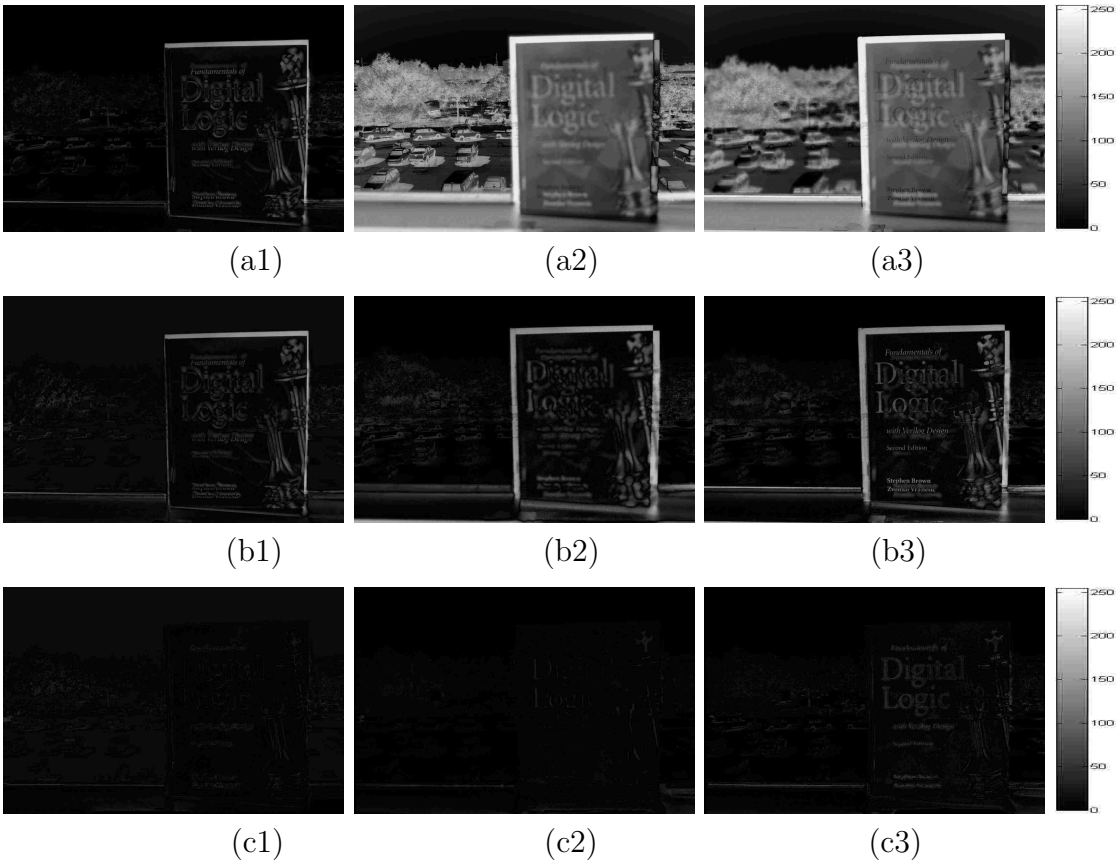


FIGURE 5.6. Absolute difference maps of intensity values between reference image and target images before and after spatial registration (Luminance channel of color images). (a1)-(a3) Difference maps between original reference image shown in Figure 5.3(a) and three target images shown in Figures 5.3(b)-5.3(d) (Mean absolute differences (MADs) are 9.4, 82.6 and 80.6, respectively); (b1)-(b3) Difference maps between reference image and target images after histogram-based IMF for photometric registration (MADs are 12.8, 18.0 and 17.3, respectively); (c1)-(c3) Difference maps after histogram-based IMF and optical flow method for spatial registration (MADs are 6.9, 3.0 and 5.1, respectively).

Figure 5.7 shows focus stacking results when each of the two dark images with exposure time of 1/1600 second is used as reference image and other three images are used as target images for focus stacking under exposure diversity. Figures 5.7(a) and 5.7(b) are two dark images with different focus regions, Figures 5.7(c1) and 5.7(c2) shows created all-in-focus image without and with reliability mask when Figure 5.7(a) is used as reference image for

focus stacking, whereas Figures 5.7(d1) and 5.7(d2) shows created all-in-focus image without and with reliability mask when Figure 5.7(b) is used as reference image for focus stacking.

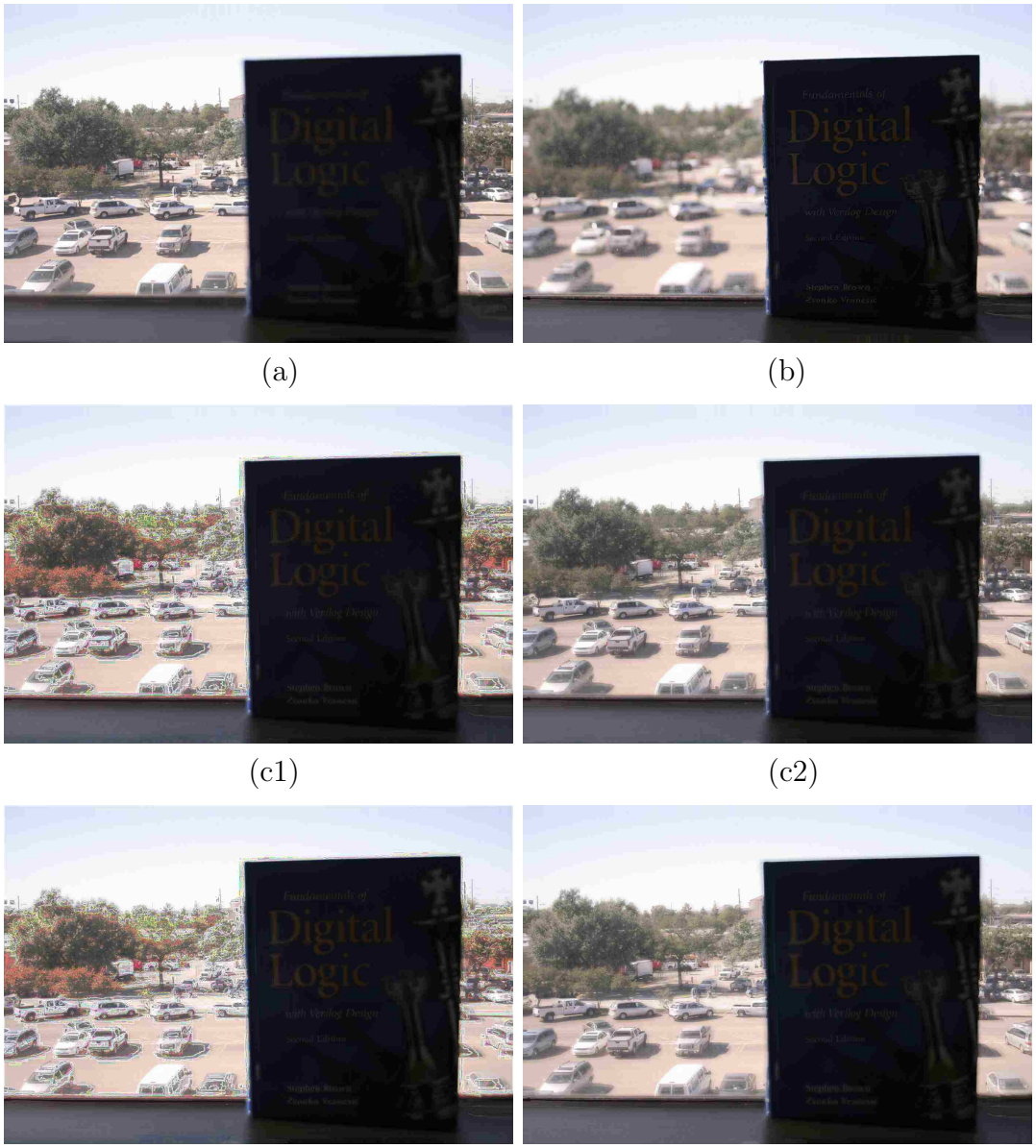


FIGURE 5.7. Focus stacking results for dark images. (a) and (b) Two dark images (shown in Figures 5.5(c) and 5.5(d)) with differently focused regions; (c1) and (c2) Focus stacked image without and with reliability check when dark image in Figure 5.5(c) is used as reference image for intensity registration and focus stacking; (d1) and (d2) Focus stacked image without and with reliability check when dark image in Figure 5.5(d) is used as reference image for intensity registration and focus stacking.

Figure 5.8 shows zoomed-in versions of Figure 5.7. Obviously, without reliability mask, the created all-in-focus image can extend depth of field in some regions; however, it also contains unwanted artifacts in some other regions. By comparison, the proposed focus stacking method with reliability mask can not only extend depth of field from differently focused and exposed images, but also eliminate artifacts introduced by possible registration errors.

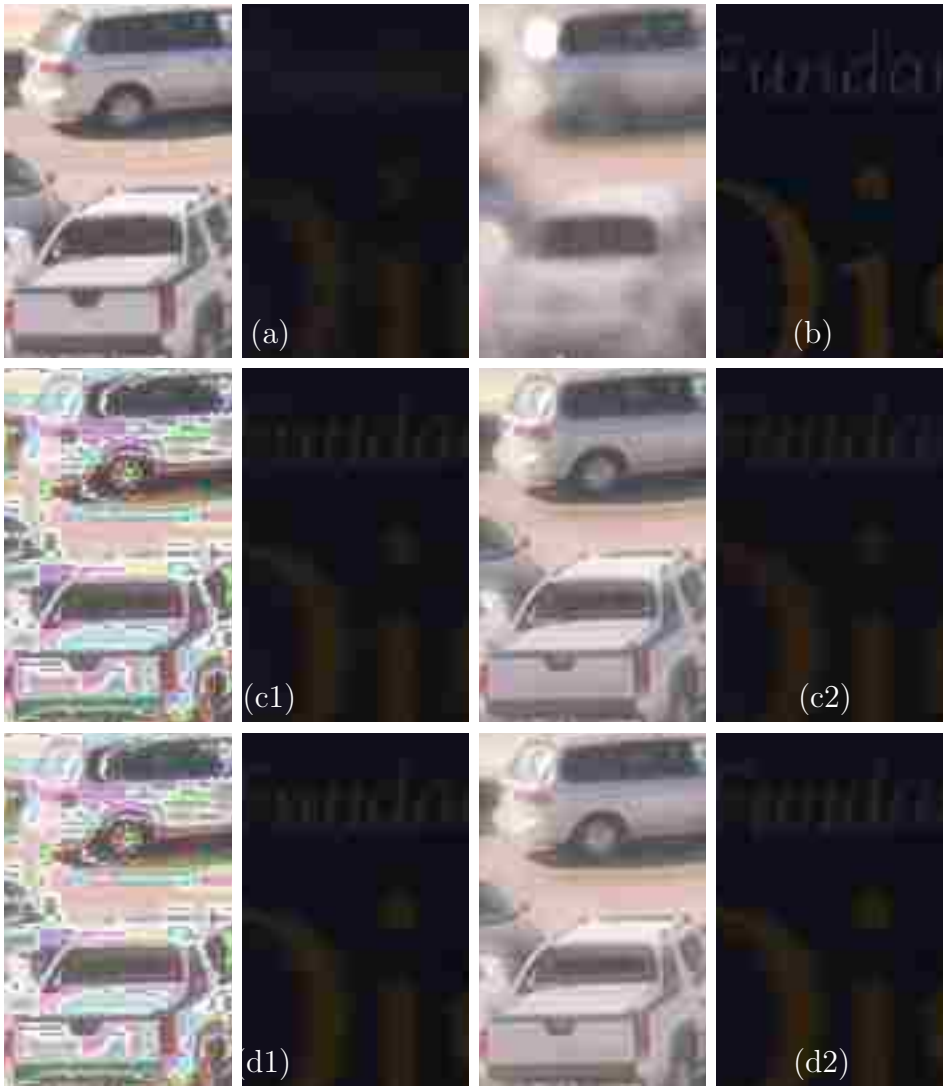


FIGURE 5.8. Zoomed-in focus stacking results for dark images. (a) and (b) Two dark images (shown in Figures 5.5(c) and 5.5(d)) with differently focused regions; (c1) and (c2) Focus stacked image without and with reliability check when dark image in Figure 5.5(c) is used as reference image for intensity registration and focus stacking; (d1) and (d2) Focus stacked image without and with reliability check when dark image in Figure 5.5(d) is used as reference image for intensity registration and focus stacking.

Figures 5.9 and 5.10 show focus stacking results when each of the two bright images with exposure time of 1/200 is used as reference image and other three images are used as target images for focus stacking under exposure diversity. Although the effectiveness of reliability mask for bright images is not as obvious as in the cases of dark images. The reliability mask still needs to be included for possible registration errors in other image samples.

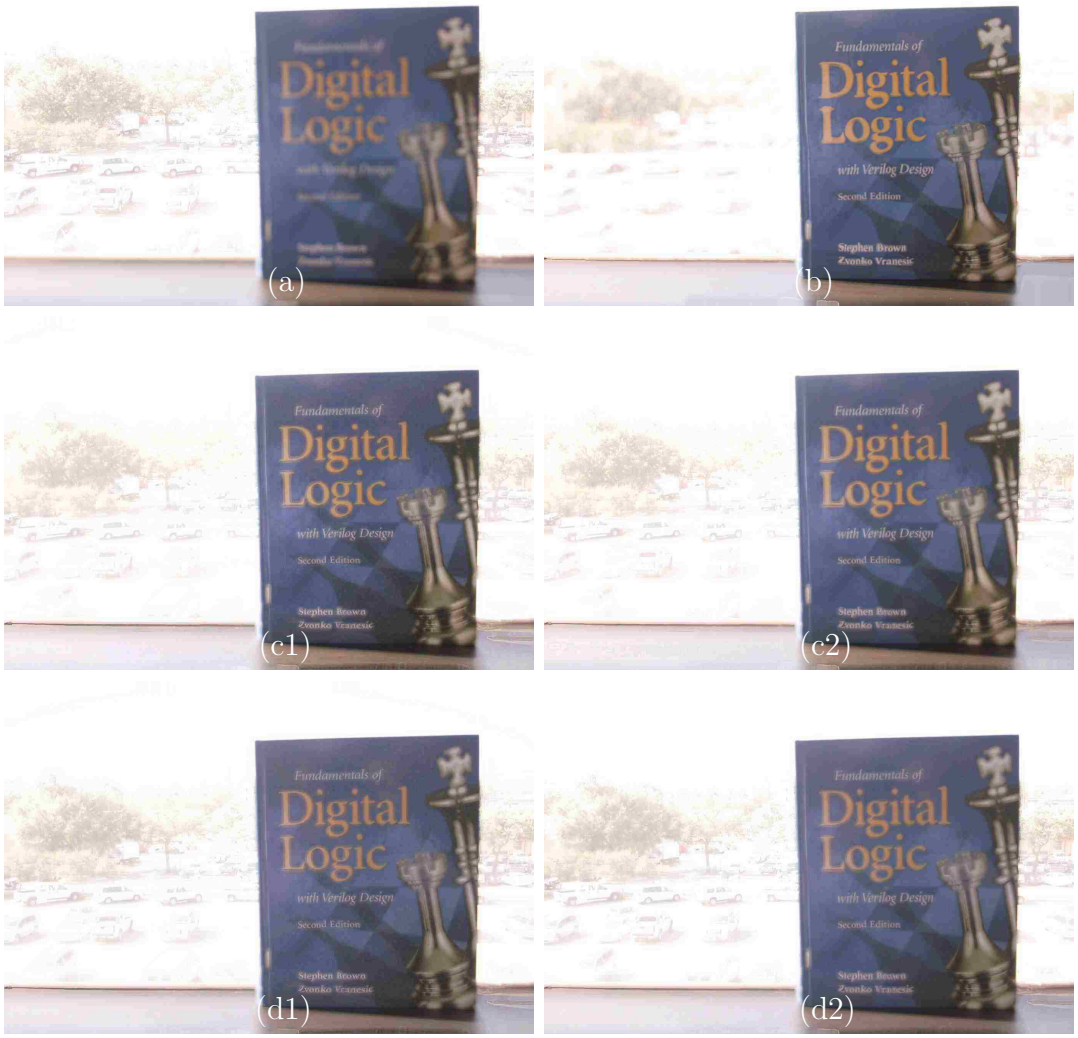


FIGURE 5.9. Focus stacking results for bright images. (a) and (b) Two bright images (shown in Figures 5.5(a) and 5.5(b)) with differently focused regions; (c1) and (c2) Focus stacked image without and with reliability check when dark image in Figure 5.5(a) is used as reference image for intensity registration and focus stacking; (d1) and (d2) Focus stacked image without and with reliability check when dark image in Figure 5.5(b) is used as reference image for intensity registration and focus stacking.

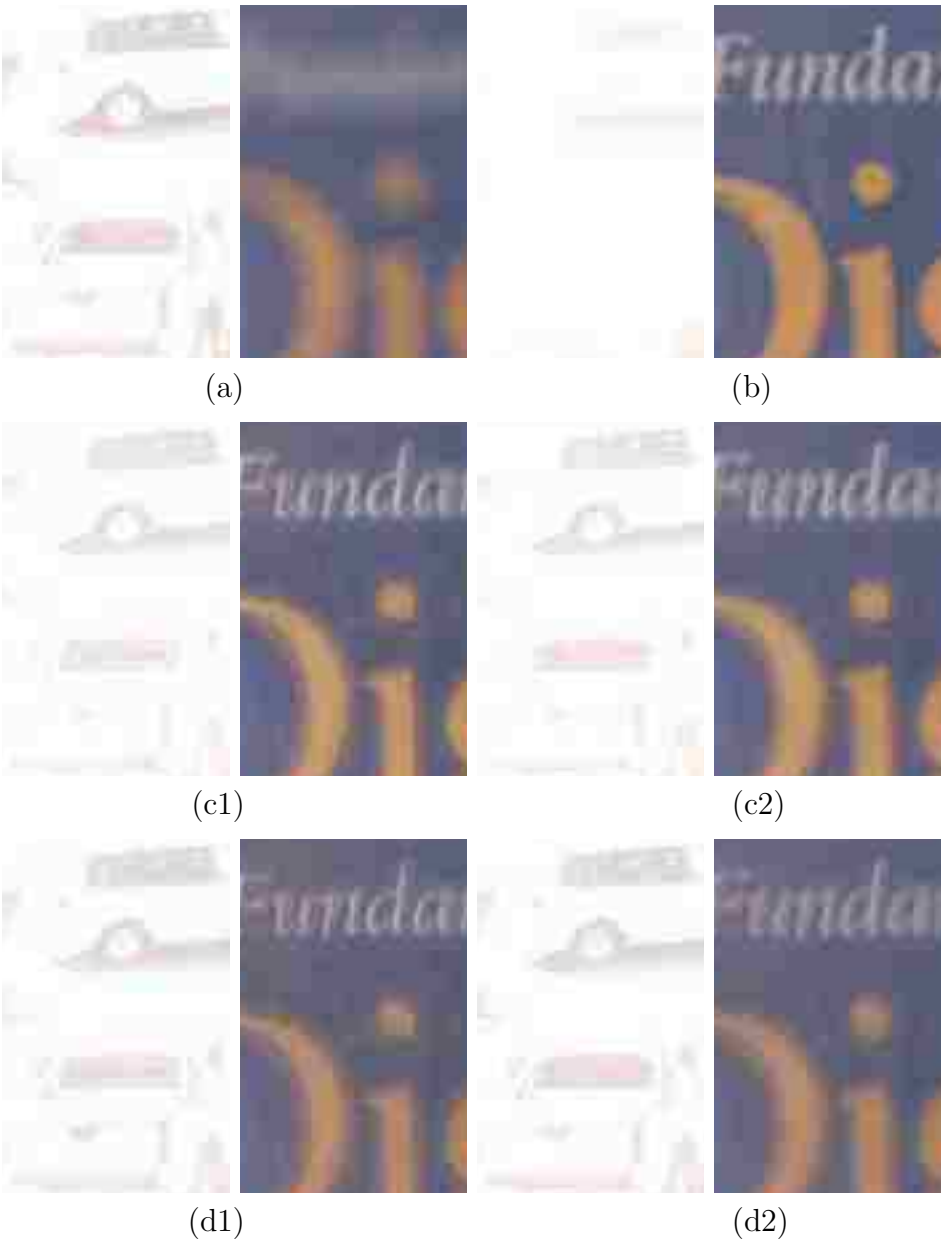


FIGURE 5.10. Zoomed-in focus stacking results for bright images. (a) and (b) Two bright images (shown in Figures 5.5(a) and 5.5(b)) with differently focused regions; (c1) and (c2) Focus stacked image without and with reliability check when dark image in Figure 5.5(a) is used as reference image for intensity registration and focus stacking; (d1) and (d2) Focus stacked image without and with reliability check when dark image in Figure 5.5(b) is used as reference image for intensity registration and focus stacking.

In the next step, the created four all-in-focus images are used for HDR radiance estimation and tone mapping. Figure 5.11 shows HDR results. Figure 5.11(a) shows HDR tone mapped image when four registered images in Figure 5.5 are used for HDR creation. Figure 5.11(b)

FIGURE 5.11. HDR results. (a1) HDR tone mapped image where four registered images in Figures 5.5(a)-5.5(d) are used for HDR creation; (b1) HDR tone mapped image where four all-in-focus images without reliability mask (shown in Figures 5.9(c1), 5.9(d1), 5.7(c1) and 5.7(d1)) are used for HDR creation; (c1) Tone mapped image where four all-in-focus images with reliability mask (shown in Figures 5.9(c2), 5.9(d2), 5.7(c2) and 5.7(d2) are used for HDR creation; (a2)-(c2) Zoomed-in versions of (a1)-(c1).

shows HDR tone mapped image when four created all-in-focus images without reliability mask shown in Figures 5.7(c1), 5.7(d1), 5.9(c1) and 5.9(d1) are used for HDR creation. Figure 5.11(c) shows HDR tone mapped image when four created all-in-focus images with reliability mask shown in Figures 5.7(c2), 5.7(d2), 5.9(c2) and 5.9(d2) are used for HDR

creation. By comparison, my proposed focus stacking under exposure diversity and HDR method can extend depth of field and dynamic range of a scene from differently exposed and focused images.

In my second experiment, I use only two images with a different combination of exposure time and focus area. Figures 5.12, 5.13 and 5.14 show experimental results of spatial registration, focus stacking under exposure diversity and HDR creation, respectively. The two input

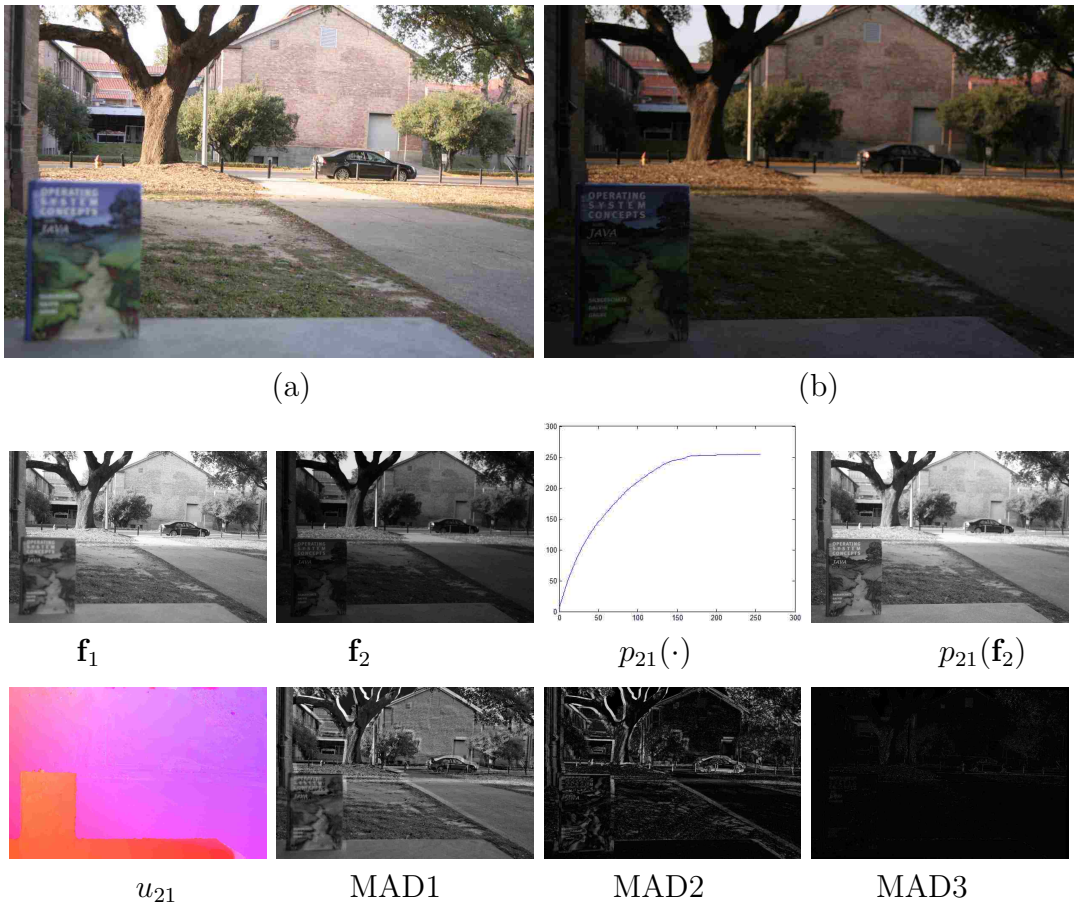


FIGURE 5.12. Spatial registration for another example. (a) and (b) Reference image and target image; (a) and (b) have different focus regions and exposure time (1/20 second, 1/33 second); MAD1: Mean absolute difference before spatial registration; MAD2: Mean absolute difference after IMF but before optical flow method; MAD3: Mean absolute difference after IMF and optical flow method for spatial registration.

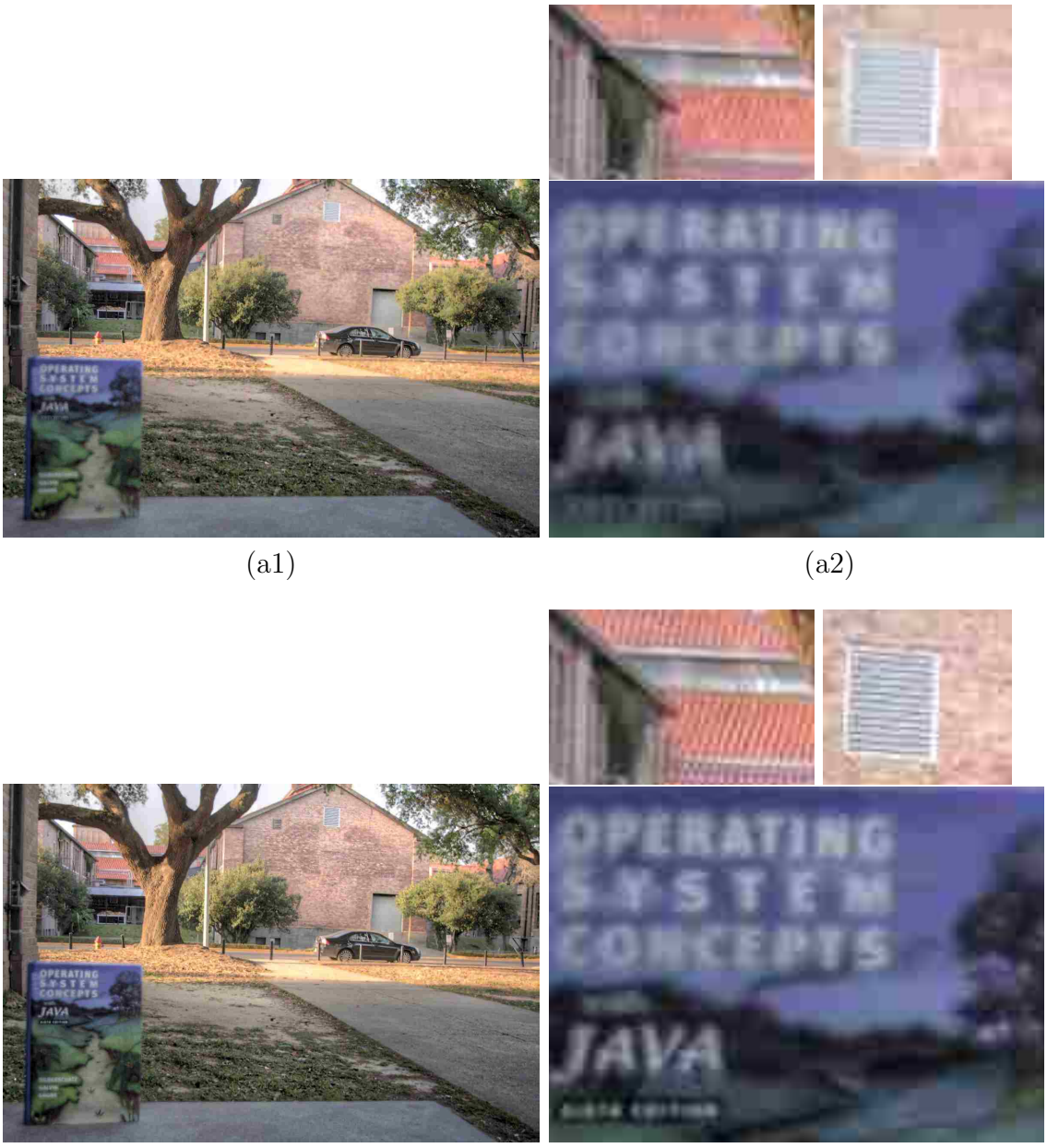
images (shown in Figure 5.12(a) and 5.12(b)) are also captured with a DSLR camera. Figure 5.12(a) has a exposure time of 1/20 second, and is far-focused, whereas the second image

in Figure 5.12(b) has a exposure time of 1/33 second, and is near-focused. The created two all-in-focus images in Figure 5.13(c1) and 5.13(d1) and HDR tone mapped image in Figure 5.14(b1) demonstrate the effectiveness of my proposed method. From the created images, we may see that our proposed focus stacking under exposure deversity extends depth of field, and subsequent HDR creation extends dynamic range of the scene. The final created image shown in Figure 5.14(b1) extends both dynamic range and depth of field of the scene.

5.4 Conclusions

In this chapter, I presented a method for extending depth of field and dynamic range from differently focused and exposed images. Spatial registration is required to get the same scene for merging differently exposed and focused images. Focus stacking under exposure diversity first uses local sharpness-based criteria to detect in-focus regions, and then performs image fusion which uses reliability mask to rule out registration error to create all-in-focus images. Lastly, HDR estimation and tone mapping operation are applied to extend the dynamic range of created all-in-focus images. Experimental results on real data show that the proposed algorithm is promising to bring this application to consumer level.

FIGURE 5.13. Focus stacking results for another example. (a) and (b) Two spatially registered images with differently focused regions and different exposure values; (c1) and (c2) Focus stacked image with reliability check when (a) is used as reference image for intensity registration and focus stacking; (d1) and (d2) Focus stacked image with reliability check when (b) is used as reference image for intensity registration and focus stacking.



(a1) (a2)

(b1) (b2)

FIGURE 5.14. HDR results for another example. (a1) HDR tone mapped image where two registered images in Figures 5.13(a) and 5.13(b) are used for HDR creation; (b1) Tone mapped image where two all-in-focus images with reliability mask (shown in Figures 5.13(c1) and 5.13(d1) are used for HDR creation; (a2)-(b2) Zoomed-in versions of (a1)-(b1).

6 Conclusions

Computational photography has the tendency to bring novel applications and experiences to industry. Apple Inc. introduced HDR photos to iPhone4 or later which enables the user to combine three separate exposures into a single HDR photo. Adobe Inc. incorporated focus stacking as new feature into Photoshop which enables the user to create all-in-focus image from multiple images with narrow depth-of-field. Also, in Max 2011 Sneak Peak, Adobe product team announced the new feature in Photoshop that allows the user to remove blurriness from digital photos caused by camera shake. Lytro Inc. brings light field camera to the common customers which allows the user to refocus the photo to either background or foreground after the photo is taken. In my dissertation, I also bring computational photography applications such as super-resolution imaging, HDR imaging and focus stacking to PC and/or Android platform in Chapter 2. The emerging market of computational photography indicates that vast investigation of image enhancement techniques in computational photography has been motivated by research teams in industry from the marketing point of view. Image quality is degraded by image formation process in traditional photography. As a consequence, image enhancement methods seek to compensate for image degradations such that it can escape the limitations of traditional photography.

Although super-resolution imaging and motion deblurring have been extensively studied in each subfield, little attention has been given to the super-resolution problem where each image is possibly degraded by large motion blur. Chapter 3 proposed a unified framework which combined the super-resolution problem and the single image deblurring problem. A novel

weight-based super-resolution reconstruction model is then formed in which each weight is associated with the motion deblurred image, indicating the contribution of the corresponding motion deblurred image in the overall cost to be minimized. Chapter 4 investigated the space-varying image deblurring problem where a new way of identifying blur kernels across the entire image is proposed. A new framework is proposed which contained three main steps under the assumptions that the local blur kernel is uniform and that blur kernels across the entire image could be clustered into distinct classes. No prior knowledge of blur kernel type or structure information about objects affected by blur kernels is required beforehand. Deblurred images from refined blur kernels will be used as candidates for final restored image by the image fusion process. Although HDR and focus stacking problems have been actively studied and even widely developed as software products as well. A unified framework of HDR and focus stacking has seldom been investigated in the computational photography field. One of main contributions of my dissertation is to propose a unified framework of extending depth-of-field and dynamic range from differently focused and exposed images in Chapter 5.

Although a variety of image enhancement methods and applications were investigated in my dissertation, there are still many issues to be addressed. My space-varying image deblurring algorithm does not handle boundary information or structure information of object, leading to artifacts around boundaries of objects. The suitable size of sliding window to get patches should be studied. In the blur kernel PSF estimation problem, I use hard thresholding to enforce sparsity of blur kernel. Actually, to robustly enforce sparsity, the nature or structure of blur kernel needs to be studied in my future work. Also, in both super-resolution problem and image deblurring problem, computational cost should be reduced for real-time applications by introducing GPU accelerators in the future work. Nowadays, multi-threaded GPU implementation can reduce the execution time for intensive computations in super-resolution and image deblurring problems. Furthermore, reduced per-frame processing time

allows the user to apply these image enhancement techniques to the real-time video streams which may be one promising direction in computational photography.

Another issue to be investigated in my future work is how to adaptively set parameters in the proposed image enhancement methods. For example, in my proposed method for extending depth of field and dynamic range of a scene, we empirically choose the threshold to reject registration errors. Also, in our proposed space-varying image deblurring method, we empirically choose a certain number of patches whose blur kernels are closer to the estimated centroid as the reliable region for the true blur kernel. In my future work, some objective methods may be provided to robustly set parameters in the proposed image enhancement methods and applications.

References

[1] Hayes, B., "Computing science: Computational photography," *American Scientist* **96**, 94–98 (April 2008).

[2] Nayar, S. K., "Computational cameras: Redefining the image," *IEEE Computer Magazine, Special Issue on Computational Photography* , 30–38 (August 2006).

[3] Raskar, R. and Tumblin, J., "Computational photography: Mastering new techniques forllenses, lighting, and sensors," in [*A K Peters, Wellesley, Massachusetts*], (2010).

[4] Levoy, M., "Light fields and computational imaging," *Computer* **39**, 46–55 (August 2006).

[5] Szeliski, R., "Computer vision: Algorithms and applications," in [*Springer*], (2010).

[6] Dowski, E. R. and Cathey, W. T., "Extended depth of field through wave-front coding," *Applied Optics* **34**, 1859–1866 (1995).

[7] Levin, A., Fergus, R., Durand, F., and Freeman, W., "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics* **26**, 1–9 (August 2007).

[8] Ng, R., Levoy, M., Bredif, M., Duval, G., Horowitz, M., and Hanrahan, P., "Light field photography with a hand-held plenoptic camera," in [*Stanford Tech Report CTSR 2005-02*], (April 2005).

[9] Debevec, P. and Malik, J., "Recovering high dynamic range radiance maps from photographs," in [*Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*], 369–378, ACM Press/Addison-Wesley Publishing Co (1997).

[10] Mitsunaga, T. and Nayar, S., "Radiometric self calibration," in [*Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*], 374–380, IEEE (1999).

[11] Mitsunaga, T. and Nayar, S., "High dynamic range imaging: Spatially varying pixel exposures," in [*Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*], 472–479, IEEE (2000).

[12] Raskar, R., Agrawal, A., and Tumblin, J., "Coded exposure photography: motion deblurring using fluttered shutter," *ACM Transactions on Graphics* **25**, 795–804 (2006).

[13] Park, S. C., Park, M. K., and Kang, M. G., "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine* **20**, 21–36 (May 2003).

[14] Farsiu, S., Robinson, M. D., Elad, M., and Milanfar, P., "Fast and robust multiple-frame super resolution," *IEEE Transaction on Image Processing* **13**, 1327–1343 (October 2004).

[15] Elad, M. and Hel-Or, Y., "A fast super-resolution reconstruction algorithm for prue translational motion and common space-invariant blur," *IEEE Transaction on Image Processing* **10**, 1187–1193 (August 2001).

[16] Mitzel, D., Pock, T., Schoenemann, T., and Cremers, D., "Video super resolution using duality based tv-l1 optical flow," in [*the 31st DAGM Symposium on Pattern Recognition*], 432–441, Springer-Verlag (2009).

[17] Mudenagudi, U., Banerjee, S., and Kalra, P. K., "Space-time super-resolution using graph-cut optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**, 995–1008 (May 2011).

[18] Su, H., Tang, L., and Wu, Y., "Spatially adaptive block-based super-resolution," *IEEE Transactions on Image Processing* **21**, 1031–1045 (March 2012).

[19] Pieper, R. J. and Korpel, A., "Image processing for extended depth of field," *Applied Optics* **22**, 1449–1453 (1983).

[20] Yeo, T., Ong, S., and Sinniah, R., "Autofocusing for tissue microscopy," *Image and Vision Computing* **11**, 629–639 (December 1993).

[21] Forster, B., Van De Ville, D., Berent, J., Sage, D., and Unser, M., "Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images," *Microscopy Research and Technique* **65**, 33–42 (September 2004).

[22] Kaneda, K., Ishida, S., Ishida, A., and Nakamae, E., "Image processing and synthesis for extended depth of field of optical microscopes," *Visual Computer* **8**, 351–360 (1992).

[23] Antunes, M., Trachtenberg, M., Thomas, G., and Shoa, T., "All-in-focus imaging using a series of images on different focal planes," *Lecture Notes Computer Science* **3656**, 174–181 (2005).

[24] Sugimoto, S. A. and Ichioka, Y., "Digital composition of images with increased depth of focus considering depth information," *Applied Optics* **24**, 2076–2080 (1985).

[25] Qian, Q., Gunturk, B. K., and Batur, A. U., "Joint focus stacking and high dynamic range imaging," in [*SPIE Electronic Imaging Conference*], SPIE (2013).

[26] Qian, Q. and Gunturk, B. K., "Super-resolution restoration of motion blurred images," in [*SPIE Electronic Imaging Conference*], SPIE (2014).

[27] Qian, Q. and Gunturk, B. K., "Space-varying blur kernel estimation and image deblurring," in [*SPIE Electronic Imaging Conference*], SPIE (2014).

[28] Milanfar, P., "Super-resolution imaging," in [*CRC Press*], (September 2010).

[29] Tai, Y. W., Tang, H. X., Brown, M. S., and Lin, S., "Detail recovery for single-image defocus blur," *IPSJ Transactions on Computer Vision and Applications* **1**, 1–10 (March 2009).

[30] Bae, S. and Durand, F., "Defocus magnification," in [*Comput. Graph. Forum*], 571–579 (2007).

[31] Tai, Y. W. and Brown, M. S., "Single image defocus map estimation using local contrast prior," in [*16th IEEE International Conference on Image Processing*], 1797–1800 (2009).

[32] Shen, C. T., Hwang, W. L., and Pei, S. C., "Spatially-varying out-of-focus image deblurring with l1-2 optimization and a guided blur map," in [*37th IEEE International Conference on Acoustics, Speech, and Signal Processing*], 1069–1072 (2012).

[33] Fergus, R., Singh, B., Hertzmann, A., and Freeman, W., "Removing camera shake from a single photograph," *ACM Transactions on Graphics* **25**, 787–794 (2006).

[34] Shan, Q., Jia, J., and Agarwala, A., "High-quality motion deblurring from a single image," *ACM Transactions on Graphics* **27**, 1–10 (August 2008).

[35] Cai, J., Ji, H., Liu, C., and Shen, Z., "Blind motion deblurring from a single image using sparse approximation," *Computer Vision and Pattern Recognition* , 104–111 (2009).

[36] Joshi, N., Szeliski, R., and Kriegman, D., "Psf estimation using sharp edge prediction," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (June 2008).

[37] Cho, S. and Lee, S., "Fast motion deblurring," *ACM Transactions on Graphics* **28**, article no. 145 (December 2009).

[38] Xu, L. and Jia, J. Y., "Two-phase kernel estimation for robust motion deblurring," in [*11th European conference on Computer vision*], 157–170, Springer-Verlag (2010).

[39] Chen, J., Yuan, L., Tang, C. K., and Quan, L., "Robust dual motion deblurring," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (June 2008).

[40] Li, F., Yu, J. Y., and Chai, J. X., "A hybrid camera for motion deblurring and depth map super-resolution," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (June 2008).

[41] Yuan, L., Sun, J., Quan, L., and Shum, H. Y., "Image deblurring with blurred/noisy image pairs," *ACM Transactions on Graphics* **26**, Article no. 1 (2007).

[42] Harmeling, S., Michael, H., and Schoelkopf, B., "Space-variant single-image blind deconvolution for removing camera shake," in [*NIPS Advances in Neural Information Processing Systems 23*], 829–837 (2010).

[43] Joshi, N., "Enhancing photographs using content-specific image priors," in [*PhD thesis, University of California, San Diego*], (2008).

[44] Whyte, O., Sivic, J., Zisserman, A., and Ponce, J., "Non-uniform deblurring for shaken images," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (2010).

[45] Xu, L., Zheng, S., and Jia, J. Y., "Unnatural l0 sparse representation for natural image deblurring," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (2013).

[46] Xu, L. and Jia, J. Y., "Depth-aware motion deblurring," in [*IEEE International Conference on Computational Photography*], (2012).

[47] Levin, A., "Blind motion deblurring using image statistics," in [*NIPS Advances in Neural Information Processing Systems*], 841–848 (2006).

[48] Liu, R., Li, Z., and Jia, J., "Image partial blur detection and classification," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (2008).

[49] Chakrabarti, A., Zickler, T., and Freeman, W. T., "Analyzing spatially-varying blur," in [*IEEE Conference on Computer Vision and Pattern Recognition*], (2010).

[50] Shan, Q., Xiong, W., and Jia, J. Y., "Rotational motion deblurring of a rigid object from a single image," in [*IEEE International Conference on Computer Vision*], 1–8 (2007).

[51] Cho, S., Matsushita, Y., and Lee, S., "Removing non-uniform motion blur from images," in [*IEEE International Conference on Computer Vision*], 1–8 (2007).

[52] Osher, S. and Rudin, L., "Feature-oriented image enhancement using shock filters," *SIAM Journal on Numerical Analysis* **27**, 919–940 (1990).

[53] Richardson, W. H., "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America* **62**, 55–59 (January 1972).

[54] Lucy, L. B., "An iterative technique for the rectification of observed distributions," *Astronomical Journal* **79**, 745–754 (January 1974).

[55] Wang, J. F., Zhang, Y., and Yin, W. T., "An efficient tvl1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM Journal on Scientific Computing* **31**, 2842–2865 (July 2009).

[56] Shepp, L. A. and Vardi, Y., "Maximum likelihood reconstruction for emission tomography," *IEEE Transactions on Medical Imaging* **1**, 745–754 (October 1982).

[57] Gunturk, B. K., "Super-resolution imaging," in [*Computational Photography: Methods and Applications, CRC Press*], (September 2010).

[58] Gharavi, H. and Mills, M., "Block-matching motion estimation algorithms-new results," *IEEE Transactions on Circuits and Systems* **37**, 649–651 (May 1990).

[59] Horn, B. K. P. and Schunck, B. G., "Determining optical flow.," *Artificial Intelligence* **17**, 185–203 (April 1981).

[60] Brox, T., Bruhn, A., Papenberg, N., and Weickert, J., "High accuracy optical flow estimation based on a theory for warping," in [*8th European Conference on Computer Vision*], **4**, 25–36, Springer (2004).

[61] Liu, C., "Beyond pixels: Exploring new representations and applications for motion analysis," in [*Doctoral Thesis*], Massachusetts Institute of Technology (2009).

[62] Schultz, R. R. and Stevenson, R. L., "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing* **5**, 996–1011 (June 1996).

[63] Hardie, R., Barnard, K., and Armstrong, E., "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing* **6**, 1621–1633 (December 1997).

[64] Liu, C. and Sun, D. Q., "A bayesian approach to adaptive video super resolution," in [*24th IEEE Conference on Computer Vision and Pattern Recognition*], **2**, 209–216 (2011).

[65] Bascle, B., Blake, A., and Zisserman, A., "Motion deblurring and super-resolution from an image sequence," in [*4th European Conference on Computer Vision*], **2**, 573–582, Springer-Verlag (1996).

[66] Tsai, R. Y. and Huang, T. S., "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing, JAI Press* (1984).

[67] Kim, S., Bose, N., and Valenzuela, H., "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Transactions on Acoustics, Speech and Signal Processing* **38**, 1013–1027 (June 1990).

[68] Stark, H. and Oskoui, P., "High-resolution image recovery from image-plane arrays, using convex projections," *Journal of the Optical Society of America A* **6**, 1715–1726 (1989).

[69] Tekalp, A. M., Ozkan, M. K., and Sezan, M. I., "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in [*Proceedings of IEEE International Conference Acoustics, Speech, and Signal Processing*], 169–172 (March 1992).

[70] Patti, A. J., Sezan, M. I., and Tekalp, A. M., "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Transactions on Image Processing* **6**, 1064–1076 (August 1997).

[71] Irani, M. and Peleg, S., "Improving resolution by image registration," *CVGIP: Graphical Models and Image Processing* **35**, 231–239 (May 1991).

[72] Zomet, A. and Peleg, S., "Super-resolution from multiple images having arbitrary mutual information," in [*Boston: Kluwer Academic Publishers*], (2001).

[73] Engl, H., Hanke, M., and Neubauer, A., "Regularization of inverse problems," in [*Dordrecht: Kluwer Academic Publishers*], (1996).

[74] Davies, E. R., "Machine vision theory, algorithms, practicalities (third edition)," in [*Elsevier*], (2005).

[75] El-Yamany, N. A. and Papamichalis, P. E., "Robust color image superresolution: an adaptive m-estimation framework," *Journal on Image and Video Processing* **2008**, 16:1–16:12 (January 2008).

[76] Patanavijit, V. and Jitapunkul, S., "A lorentzian stochastic estimation for a robust iterative multiframe super-resolution reconstruction with lorentzian-tikhonov regularization," *EURASIP Journal on Advances in Signal Processing* **2007**, 21–21 (June 2007).

[77] Ward, G., "Fast, robust image registration for compositing high dynamic range photographs from handheld exposures," *Journal of Graphics Tools* **8**, 17–30 (2003).

[78] Mann, S. and Picard, R. W., "On being undigital with digital cameras: Extending dynamic range by combining differently exposed pictures," in [*ISTs 48th Annual Conference*], 422–428 (1995).

[79] Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J., "Photographic tone reproduction for digital images," *ACM Transactions on Graphics* **21**, 267–276 (July 2002).

[80] Fattal, R., Lischinski, D., and Werman, M., "Gradient domain high dynamic range compression," *ACM Transactions on Graphics* **21**, 249–256 (July 2002).

[81] Pattanaik, S. N., Tumblin, J., Yee, H., and Greenberg, D. P., "Time-dependent visual adaptation for fast realistic image display," in [*Proceedings of the 27th annual conference on Computer graphics and interactive techniques*], 47–54 (2000).

[82] Durand, F. and Dorsey, J., "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Transactions on Graphics* **21**, 257–266 (July 2002).

[83] Kang, S. and et al., "High dynamic range video," *ACM Transactions on Graphics* **22**, 319–325 (July 2003).

[84] Cossairt, O., Zhou, C. Y., and Nayar, S., "Diffusion coded photography for extended depth of field," *ACM Transactions on Graphics* **29**, 1–10 (2010).

[85] Yamaguchi, T., Fukuda, H., Furukawa, R., Kawasaki, H., and Sturm, P., "Video deblurring and super-resolution technique for multiple moving objects," in [*Proceedings of the 10th Asian conference on Computer vision - Volume Part IV*], 127–140 (2010).

[86] Sroubek, F., Cristobal, G., and Flusser, J., "A unified approach to superresolution and multichannel blind deconvolution," *IEEE Transactions on Image Processing* **16**, 2322–2332 (September 2007).

[87] MacKay, D., "An example inference task: Clustering," in [*Information Theory, Inference and Learning Algorithms*], 284–292, Cambridge University Press (2003).

[88] Hossain, I. and Gunturk, B. K., "High dynamic range imaging of non-static scenes," in [*SPIE Electronic Imaging Conference*], **7876**, SPIE (2011).

[89] Grossberg, M. D. and Nayar, S. K., "Determining the camera response from images: what is knowable?," *IEEE Transanctions Pattern Analysis and Machine Intelligence* **25**, 1455–1467 (November 2003).

[90] Gevrekci, M. and Gunturk, B. K., "Superresolution under photometric diversity of images," *EURASIP Journal on Advances in Signal Processing* **2007** (January 2007).

Vita

Qinchun Qian was born in Yancheng, Jiangsu, China in 1982. He finished his undergraduate study in 2006, with a Bachelor degree in Information and Computing Science from School of Mathematics in Shandong University, China. In 2009, he got a M.E. degree in Control Theory and Control Engineering from Nankai University, China. In Sep, 2009 he came to Louisiana State University to pursue graduate studies in Division of Electrical and Computer Engineering, School of Electrical Engineering and Computer Science (EECS). He obtained a M.S. degree in Electrical Engineering in 2011, LSU. He is currently a candidate for the degree of Doctor of Philosophy in Division of Electrical and Computer Engineering, School of Electrical Engineering and Computer Science.